PC INTERFACED ATTENDANCE SYSTEM WITH SMS CAPABILITIES

by

Mark Anthony R. Chen

A Design Report Submitted to the School of Electrical Engineering, Electronics and Communication Engineering, and Computer Engineering in Fulfilment of the Requirements for the Degree

Bachelor of Science in Computer Engineering

Mapua Institute of Technology
March 2012

Approval Sheet

Mapúa Institute of Technology
School of EECE

This is to certify that we have supervised the preparation of and read the design report prepared by **Mark Anthony R. Chen** entitled **PC Interfaced Attendance System With SMS Capabilities** and that the said report has been submitted for the final examination by the Oral Examinations Committee.

_____

Engr. Maribelle D. Pabiania
Design Adviser

As members of the Oral Examination Committee, we certify that we have examined this design report presented before the committee on **February 22, 2012**, and hereby recommended that it be accepted as fulfilment of the design requirement for the degree in **Bachelor of Science in Computer Engineering.**

_____

Engr. Gorgonio C. Vallestero II
Panel Member

_____

Engr. Jumelyn L. Torres
Panel Member

_____

Engr. Dionis A. Padilla
Panel Member

This design report is hereby approved and accepted by the School of Electrical Engineering Electronics and Communication Engineering, and Computer Engineering as fulfilment of the design requirement for the degree in **Bachelor of Science in Computer Engineering.**

_____

Dr. Felicito S. Caluyo
Dean, School of EE-ECE-CoE

ACKNOWLEDGMENT

First of all, I would like to thank Almighty God for giving His guidance to finish this project design and also, for giving me enough strength to face all the hardships that I had encountered during its completion.

I would like to extend my deepest gratitude to my parents for supporting me in terms of the finances and their moral support to finish this project design.

I would also like to thank all my friends and classmates for giving me support whenever I needed it and for helping me by giving brilliant ideas on how I will make the design better. And to my Adviser, Engr. Maribelle D. Pabiania for helping me whenever I need supervision.

And lastly, I would like to thank Engr. Noel B Linsangan and Engr. Lilibeth Mendoza for giving me another chance on to finishing this project and sharing their knowledge to their students.

TABLE OF CONTENTS

TITLE PAGE i

APPROVAL SHEET ii

AKNOWLEDGEMENT iii

TABLE OF CONTENTS iv

LIST OF FIGURES vi

LIST OF TABLES vii

ABSTRACT viii

Chapter 1: DESIGN BACKGROUND AND INTRODUCTION 1

- Background 1
- Statement of the Problem 2
- Objective of the Design 3
- Significance and Impact of the Design 4
- Scope and Delimitation 4
- Definition of Terms 6

Chapter 2: REVIEW OF RELATED LITERATURE AND STUDIES 7

- The Embedded transmitters in the RFID wristband 7
- Mifare Classis 8
- Computerized Parking Lot Monitor using ID Barcode 8
- ZigBee Based Active RFID System 9
- MCU Coin Operated Textbooth 10
- Mobile information network between students and professors 11

Chapter 3: DESIGN PROCEDURES 12

1. Construct the Problem 14
2. Determine the Objectives 14
3. Gather the Required Data 14
4. Relevance of the Data 14
5. Sufficient Information for Design 14
6. Construct of the Prototype 15
7. Testing Device Functionality (Hardware and Software) 15
8. Final Observation and Analysis 15
9. Formulation of Conclusion and Recommendation 15
10. Hardware Development 16
11. Software Development 24

Chapter 4: TESTING, PRESENTATION, AND INTERPRETATION OF DATA 26

RFID card reader Test 26
SMS Sending Test 28
Automated Attendance Test 30

Chapter 5: CONCLUSION AND RECOMMENDATION32

Conclusion 33
Recommendation 35

BIBLIOGRAPHY 36

Appendix A: User's Manual 37
Appendix B: Prototype Pictures 46
Appendix C: Program Listing 50
Appendix D: Data Sheets 71

LIST OF FIGURES

Figure 3.1: Design Procedure Flow Chart 13

Figure 3.2: General Flow of Design 17

Figure 3.3: RFID MODULE 19

Figure 3.4: Schematic Diagram 20

Figure 3.5: The Flow of Operation of the GSM Module 22

Figure 3.6: GSM Module Schematic Diagram 23

Figure 3.7: Program Flow Chart 24

LIST OF TABLES

Table 3.1: List of Materials 18

Table 4.1: RFID Card Reader Test 27

Table 4.2: SMS Sending Test 29

Table 4.3: Automated Attendance System 31

ABSTRACT

A PC based student login system called PC Interfaced Attendance System with SMS Capabilities was designed to enable parents to track their child's entry to the campus through an SMS. The system is composed of an RFID module and a GSM module interfaced to a computer. An RFID tag is swiped to the sensor and the students profile is displayed on the monitor of a personal computer. The database then search to authenticate and validate the entry of the student which serves as his attendance. Once the student failed to swipe the RFID Module, a text message is sent to the concerned parent 20 minutes before the class ends. The text message contains a notification that their child failed to attend his class and through this device, parents are informed of their child entering the school. A queue system is employed to organize sending of the message and to avoid congestion in the GSM module inside the device.

Keywords: SMS, RFID Module, GSM Module

Chapter 1
DESIGN BACKGROUND AND INTRODUCTION

Education is an essential legacy of every parent to their children where most parents want their children to have a better education. In return, students are expected to repay their parents by having good grades. To ensure this, they should at least be monitored if they really go to the campus and with this implication, the proponent designed and created a way to give parents the information they want regarding their child's school attendance, providing them the necessary information through the use of GSM technology where the student's guardians receive an SMS whenever the student failed to log in to the classroom.

Background

The strong desire to search for new technologies led to the development of new innovations and inventions that are beneficial. Man has successfully achieved all his goals but he has never been satisfied so he continues to seek for new technologies that will not only make life easier but also improve billions of lives around the planet.

One huge factor for this is the convenience where people can use these different technologies in their daily lives. One instance is a typical classroom where attendance checking is still done manually. Oftentimes, human error is

present in attendance checking. Another similar situation is that some students do receive their daily allowance but they really do not attend their classes. In a regular basis, parents and guardians usually do not have enough time to go to schools to check on their child. This problem is common to parents since they are not aware of it. The proponent designed a possible way to address these problems through automation of class attendance with Short Messaging Service (SMS) that can inform parents and guardians when a student misses a class.

The system consisted of three major components: the Smart Card module or known as the RFID module, the GSM module, and the software module. The RFID module accepted the input information. The GSM module dealt with the SMS capabilities of the system, which was responsible for the automatic sending of SMS to the corresponding parent or guardian of the student who uses the system. And the software module will be the part of the system that handles information processing.

Statement of the Problem

General Problem

A typical classroom attendance checking is still done manually which is oftentimes the cause of human error. In addition, some professors find it difficult to do, since it is time consuming especially when the class has a huge number of students.

Specific Problems

The specific problems of the design is gaining access to the latest database with the use of RFID module interfaced in a computer and implementing RFID and SMS technology to perform specific task as attendance system.

Objective of the Design

General Objective

The general objective of the design is to automate attendance checking in a typical classroom and inform parent or guardian if the child has logged in his respective classroom for a specific course schedule through SMS.

Specific Objective

The specific objectives of the design were to integrate an RFID module in a system that acts as a sensor for the student's identification in the GSM module and to create a software that combines and controls both in doing the specific task.

Significance & Impact of the Design

This study aimed to create a working design of the system, that enhances the old system i.e. manual checking of attendance of students who are actually inside the classroom. Not only, that it can save time but also very useful for parents who want to keep track their children's attendance in school. SMS is one of the simplest ways of communication, the proponent applied this technology in creating the design. This design has a big market potential since it can be very useful to professors, parents and the students.

Furthermore, serves as a basis for other designers who are planning to incorporate SMS into to their system.

Scope and Delimitation

The sending of SMS was done by the system automatically based on the actual time spent in the class. This was the only time when parents began receiving text messages which took only a few minutes. The system used the queue system which means that messages sent were on FIFO (First IN First Out) basis.

This was given to chosen parents/guardians. They received text messages but sending reply was not possible.

For testing, prepaid cards were not used because once it runs out of credit the queued SMS messages will not be sent. The design was not also designed to handle two tags logging at the same time or if the tag was not

properly tapped. It can register or read the tag within a limited distant of about 2 centimetres far. In case a different tag is tapped on the reader and that tag is not registered in the computers data base, the tag will not work, unless it is the same kind of tag with 48 series. It will only show its ID number in the screen but will not be recognised or added in the computer's database.

The user should also make sure that there's a good signal in the area where the device is to be used because it might results tp less efficiency and failure.

The student will remain under the absent status not until he is able to login during the scheduled time of his class. So students should always make sure that the reader reads and saves their data into the computer to make sure he is present. Once the tag is tapped, the information will appear on the screen which signifies that the reader was able to read his ID tag. This changes, if another student tapped his ID. Remember only one time tapping is allowed. It's also important the best is to wait at least 3 seconds after the first student has tapped his ID before tapping again even though quick tapping is applicable. In case failed to tap, it is better to wait few seconds before he can once again tapped his ID.

There is also a display for the list of students enrolled on the said class which serves as a notification for the student if he was able to tap correctly. The absent sent can only be determined after the end of class wherein the students name will automatically be displayed below. The processing of tests shall vary

widely on the processing speed of the computer where the prototype is connected to. The message length won't exceed 160 characters.

Definition of Terms

RFID	An automatic identification method, relying on storing remotely retrieving data using devices called RFID tags or transponders.
RFID TAGS	An object that can be attached to or incorporated into a product, animal or person for the purpose of identification using radio waves. It is also compose of silicon chips and antennas.
GSM	Global System for Mobile Communications. It is the second digital technology originally developed for Europe but now has in excess of 71 percent of the world market.
GSM Module	It is responsible for the transmission and reception of encoded and decoded messages.
Chip	A small silicon device where information is stored.
SMS	Short Messaging Service. It is a service available on most digital phones or other devices that permits sending of messages.
CPU	Central Processing Unit also known as Processor, it is the brain of the computer where most calculations and processes of data take place.
Software	This are the programs that will enable computer to perform the specific task.
Database	A collection of data arranged for ease and speed of search and retrieval.
Subscriber	Alternative term for parents or guardians of the students who availed the student monitoring system

Chapter 2

Review of Related Literature

In order to obtain better design ideas and innovations, the designers made use of some of the existing devices and studies related to the design prototype. Due to the uniqueness of the device, the designers have limited resources in accumulating related information. Tech magazines and books are the best sources of information that provided the designers the latest innovations utilizing the design prototype, PC Interfaced Attendance System With SMS Capabilities. The following are the existing devices and studies that have been found.

The project design was primarily concentrated on students passing through the classrooms entrance wherein they're required to tap their ID. The proponent found similarities in the same application required to the RFID technology. One was in the Pima County Jail in Texas wherein inmates wear RFID wristbands. In the jail, a wide area receiver was installed to monitor each inmate anywhere they go with the prisons premises.

For the design application, the embedded transmitters in the wristband be embedded in the ID Cards of the students as well. The difference was that only limited scanners was installed. The main function of the passive RFID tag was to carry the unique code which was needed as an input for the device.

Another good application is the Mifare Classis, which has a secure memory chip/card called smart card. This kind of technology is specially designed for payment system wherein RFID is also used. This tag has contactless communication speed up to 106 KHz and uses very strong encryption techniques. The data is protected, making it impossible to copy or modify the content of the Mifare Classic family to tags without the correct keys(s) when it is protected. As a result Mifare becomes ideal for e-money applications, secure access, data storage and fast data collections systems. It is not only limited to these application. Printed antenna technology makes finding very thin and low cost Mifare tags (e.g. labels and stickers) possible, thus ending the field of RFID applications.

Along with these examples is Mapua's Cardinal Plus. This is a two piece plastic card with a microchip and antenna inserted in between. It is used not only for identification purposes but also for storing records such as students log, guidance records, and schedules.

Tristan Calasanz (2003), directed a similar application which is the Computerized Parking Lot Monitoring System using ID Barcode. The system monitors how many cars are parked in a given parking lot while recording the duration of each automobile's stay as well as information regarding the current owner of the car(ID number, year level, student/faculty status, parking privileges). This is achieved by placing barcode scanners in front of the computer

controlled barriers at both the entrance and exit of a specific parking lot. The barcode scanner reads the personal ID of the student or faculty driving the vehicle and will crosscheck the scanned ID numbers against the database of valid users of the said parking lot. If there's no hold order against the person, then the bar is raised and the car is allowed to park. The same system applies to cars exiting the parking lot. The proponent used same concept, but in a different scenario and on a much simpler application, which is for checking the students' attendance only. An innovation on the design has also been made, instead of using barcode scanner the proponent used RFID reader with SMS capability. Every time the student taps on the reader, the student information is saved in the computers database and that will serve as the basis for class attendance.

Since RFID systems are now being assimilated into many business processes. The use of active RFID, with the onboard power sources in transponders , provides a more robust system in term of reading range and directivity compared sensitivity. The only drawback of active RFID is the need for a power source due to the utilization of active circuits. One example of this is the ZBAR, the implementation of ZigBee Based Active RFID System. The project aims to developed active RFID using EZZBMO1 modules for the reader and tags. The reader is able to collect tag ID's from multiple active tags. For the portable reader, the ZigBee is interfaced with BenQ M23 GSM module and this is the same

module that the proponent used for the design. The only difference for the ZBAR project doesn't used GPRS capabilities. ZBAR is used for individual vehicle velocity detection, traffic condition sensing like detecting traffic violations.

While for SMS technology, the MCU Coin Operated Text Booth created by a group headed by Calupitan et. al (2006) is one good application. The group used PIC microcontroller as the processing unit of the system which is connected to an internal Nokia cellular phone. A keyboard was used to gather user input, while for the output can be viewed by the operator/user through an LCD screen. The communications part of the system is performed by the NOKIA 3310 Cellular Phone unit. The Nokia 3310 is interfaced to the system using the F-bus protocol developed for mobile phones. The system accepts several coin denominations such as both 1 peso and 5 peso coins. It segregates the coins inserted by users. A change dispensing mechanism is also integrated to allow people who don't have the exact amount to send text messages and still get the change for the remaining amount. When the user inserts coins in the coin operator, the design asks the user to enter the recipient's cellphone number. Upon completion of the recipients' cellphone number, the user is asked to enter the text messages. After entering the message, the message is sent to the recipient through the cellular phone. If the user still wants to send text messages and there are still credits left, the cycle's repeats. Otherwise the change will be dispensing.

According to Ngo (2009), one of the latest related topics about monitoring attendance is the one in Japan with the use of iPhone at Aoyama Gakuin

University, which is located outside Tokyo in Sagamihara city. He has mentioned that a new testing program have been developed wherein 550 first year and second year students were given Apple, Inc.'s popular iPhone. He stated that the school's iPhones are designed to create a "mobile information network between students and professors. This devices are also an easy way for the teachers to take class attendance. For instance, when students enter the room, they will type in their ID number and a class number into an iPhone application, rather than writing their name on a sheet. Then the application uses GPS location information to check where students are logging in from. The good thing is the school is expected to pay for the iPhones' data bill, so students wouldn't incur any fees. Now this technology, they hope to develop a classroom where students and teachers can discuss various topics. Perhaps the device can also be used develop a classroom where students and teachers can discuss various topics.

Chapter 3

Design Procedures

This chapter shows a detailed procedures used in developing the design which enables in achieving the corresponding functions and objectives. Different tools in analysing the flow of data and instructions in a logical manner such as the Flow Chart, Schematic Diagram, and the Block Diagrams were shown to give readers an idea on how the design will be created.

The approaches applied in analysing both the software and hardware parts of this design have similarities such that both use flowchart to represent the process flow. Moreover, the designers used previous studies and published journals to justify and support all information based on the said presentation. The designers were able to evaluate the applications of every component necessary to describe the process of the design.

In Figure 3.1 shows the flow chart on how the proponent designed the system and how it treated the unexpected errors and modifications.

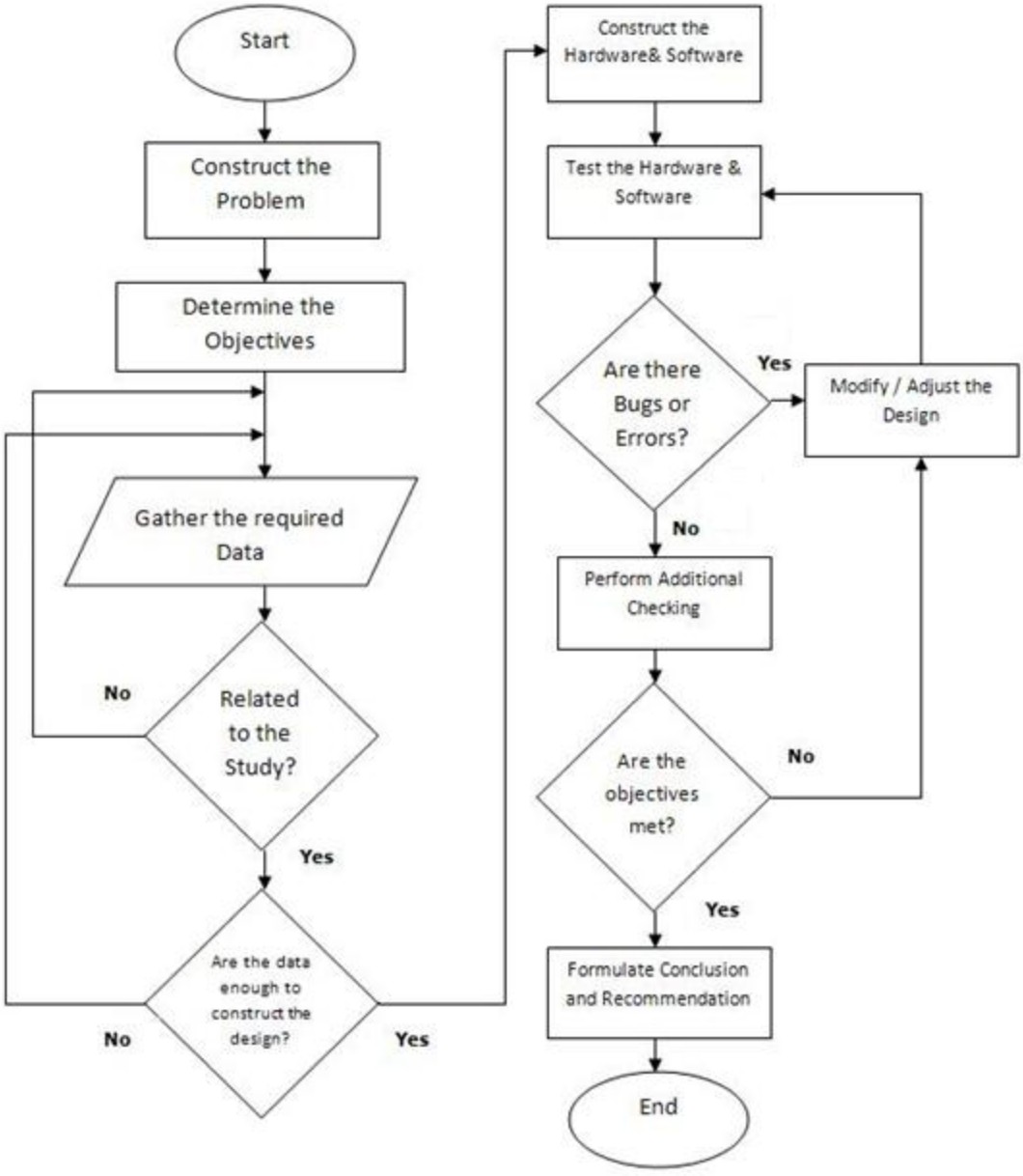


Figure 3.1: Design Procedure Flow Chart

1. Construct the Problem

   The first steps are to define and to analyse the problem, is stated in Chapter 1 of this study.

2. Determine the Objectives

   Based on the constructed problem from the first procedure, the proponent was able to determine the general and the specify objective, which are to provide the same logging system with the existing system the school has, and integrate a monitoring function for the parents to get as real-time as possible alerts via SMS from the institute.

3. Gather the Required Data

   The data gathered from the related studies helped in providing good information on the subject before the proponent could come up with a design. Online journals, previous studies and textbooks were just some of the sources of information the proponent need for the subject matter.

4. Relevance of Data

   The proponent set aside all the relevant gathered information and data for constructing the design.

5. Sufficient Information for the Design

   The information and data gathered are checked to start constructing the.

6. Construction of the Prototype

    This is the step by step guide for constructing the final prototype based from the previous gathered data's and information. This also includes the designing of schematic diagrams, programming, and the actual construction of the prototype.

7. Testing Device Functionality (Hardware and Software)

    The proponent checked and verified all possible errors by testing. This includes several kinds of testing particularly for the Sensitivity of the RFID, GSM module, and software application like its database.

8. Final Observation and Analysis

    A series of testing and checking had been performed to make sure that the device is 100% working perfectly.

9. Formulation of Conclusion and Recommendation

    Conclusion was drawn and recommendations are also made to further improve the said device.

Hardware Development

The design provides a logging system that integrates a monitoring function for the parents to get alerts via SMS. When the student user taps his identification card in the RFID module, the software will check the record of the student in the database and as an output it will automatically reflect on the professors' computer that will serve as his attendance. Then the software must be able to send SMS to the subscribed parents/guardian whose number is recorded with respect to the student database. Some record would be saved to a data pool in the data base since on SMS sending in the system requires at least 30 seconds of processing. Time after time the system will be checking for the tapping that occurs.

The 3 main components of the design are: the Smart Card Module or the RFID module, Software Module and GSM Module. The RFID Module is part of the main prototype wherein student's taps there RFID tag on it. Then Software Module processes the encrypted data from the students ID and as a result the student's information is being registered. The Software Module is also the one responsible in sending signal to the GSM module. As the GSM module receives the go signal, it will automatically send SMS to the parents. The general flow of the design is shown in Figure 3.2.

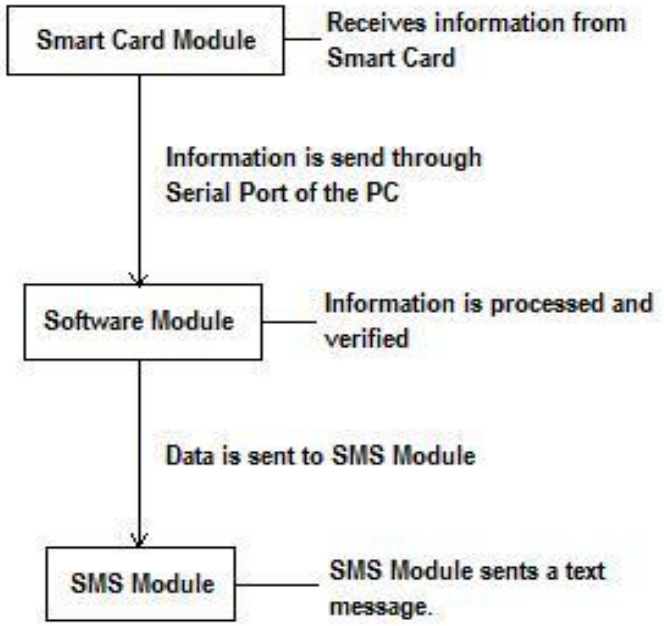

Smart Card Module
Receives information from Smart Card
Information is send through Serial Port of the PC
Software Module
Information is processed and verified
Data is sent to SMS Module
SMS Module
SMS Module sents a text message.

Figure 3.2: General Flow of Design

These are the materials used in constructing the prototype design which must comply with other components required specification as shown in Table 3.1

Components	Quantity
GSM w/ GPRS modem	1
RFID Card Reader Serial Radio freq, ID reader sensor	1
ISO RFID Card, Unique Rectangular RFID transponder tag	10
Black Casing Medium	1
Solid Wire	4ft
Voltage Regulator	1

9pin RS232 Female connector	2
Push on Round Button	1
PCB 4x6 w/ developer	1
FeCl	1
Iron Wire	1
RS232 to USB Converter	2
2n – 3906	1
Resistor	3
Transformer(power supply)	1

Table 3.1: List of Materials

RFID MODULE

The RFID module is the main source of input. It is composed of an RFID Card Reader and an RS232 to USB converter which is used to connect to the computer. The overview of or the RFID module is shown on the Figure 3.3. For its interfacing device, it is also the one in Figure 3.4 Schematic Diagram.

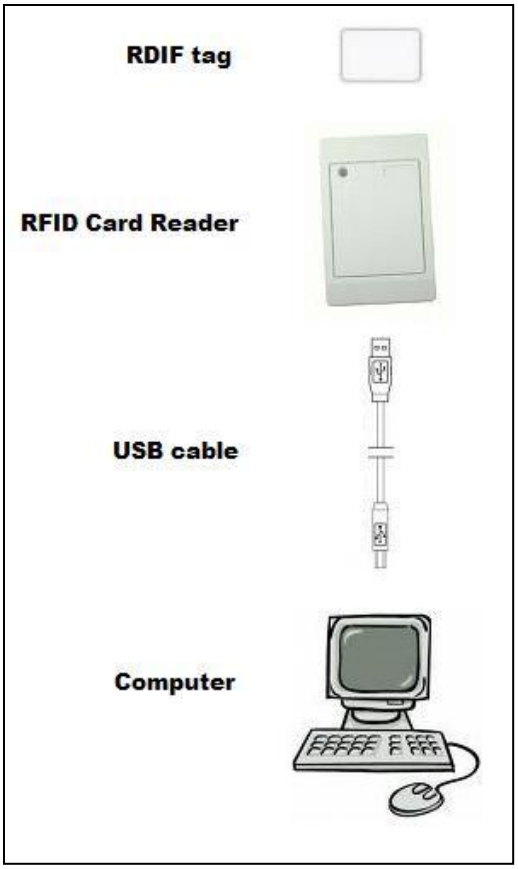


Figure 3.3: RFID MODULE

Hardware Interfacing Tool

In order to connect the computer system to the different modules, an interfacing object must bind the two (i.e. Data cable, Bluetooth, etc.). Lots are commercially available. For this project, devices that act as the interfacing object for the different modules and a computer designed to function as the driver to be able to use the internal modem of the communicating device. A good example is the design of the schematic diagram shown in Figure 3.4 which interfaces the RFID module to a PC serial port connection.

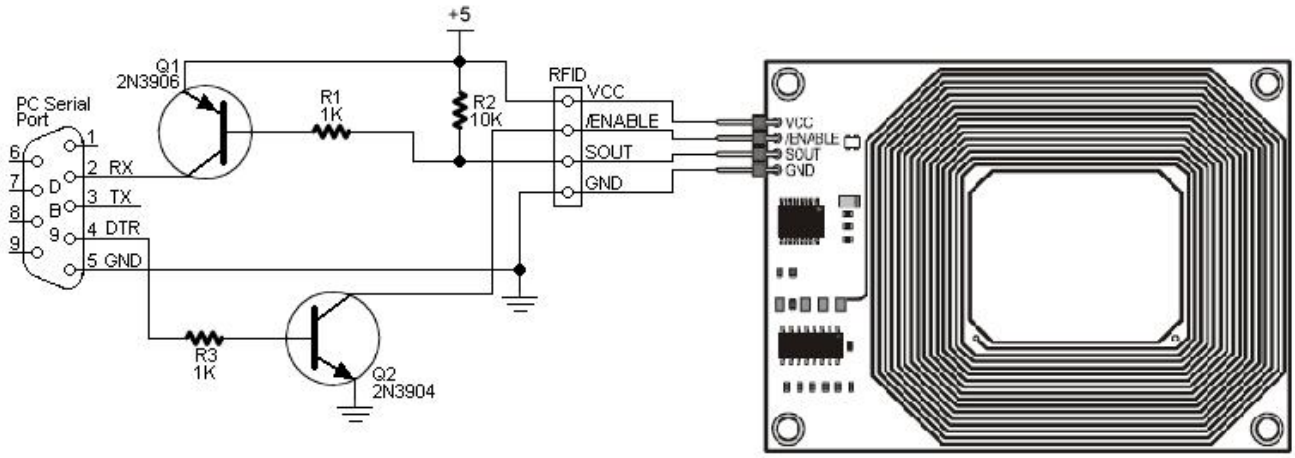


Figure 3.4: Schematic Diagram

On the RS232 female port, Pin 4 serves as the 12V input source. Basically, the overall system requires only 4.5V of input for the SIM900D and 62xx Chip; for that, the proponent used 5V voltage regulator (7805) to get the desired input rate. The 62xx chip drives the modem of the cellular phone to be used in sending commands. This enables the RX and TX(receiver and transmitter) modem ports. 62xx chip send signal over and over again to maintain the cellular phone modem connection. Once the modem is ready, signal are directly thrown to SIM900D and for the command invoking. Modem then interprets the command that has been executed from the computer to the cellular phone and it will responds on the specific commands then throw a response back to the computer interpretation.

GSM MODULE

The GSM module is developed using a GSM modem or a mobile phone. The GSM Module is connected to the PC through RS-232 or the Serial Port. To send SMS messages, first place a valid SIM card from a wireless carrier into a mobile phone or GSM/GPRS modem, which is then connected to a computer.

After connecting a mobile phone or GSM/GPRS modem to a computer, you can control the mobile phone or GSM/GPRS modem by sending instructions to it. The instructions used for controlling the mobile phone or GSM/GPRS modem are called AT commands. AT commands are the programming language used for the HyperTerminal of a computer. While a HyperTerminal is an application you can use in order to connect your computer to other remote systems.

For the proponents design, the HyperTerminal will be used to connect to mobile phone or GSM modem. The flow of operations will be:

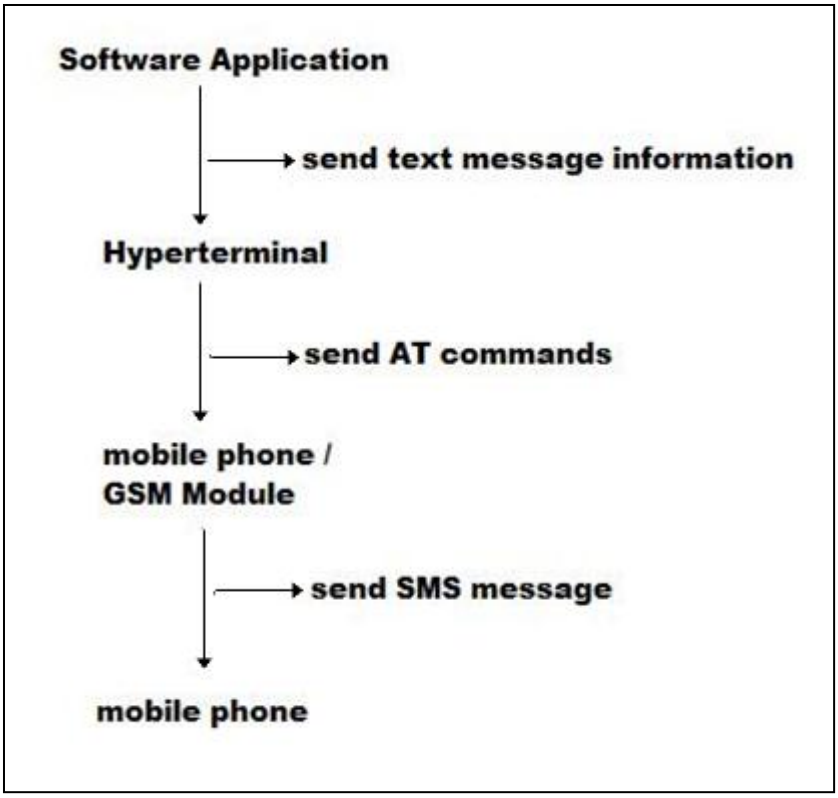


Figure 3.5: The Flow of Operation of the GSM Module

Upon receiving the text message information, the software application accesses the HyperTerminal of the computer and converts the text message information to AT commands. Then the software application makes the HyperTerminal send the AT commands to the GSM modem/mobile phone. Finally, the GSM modem/mobile phones interprets the AT commands then sends an SMS message. The designed Schematic Diagram for GSM module is in Figure 3.6

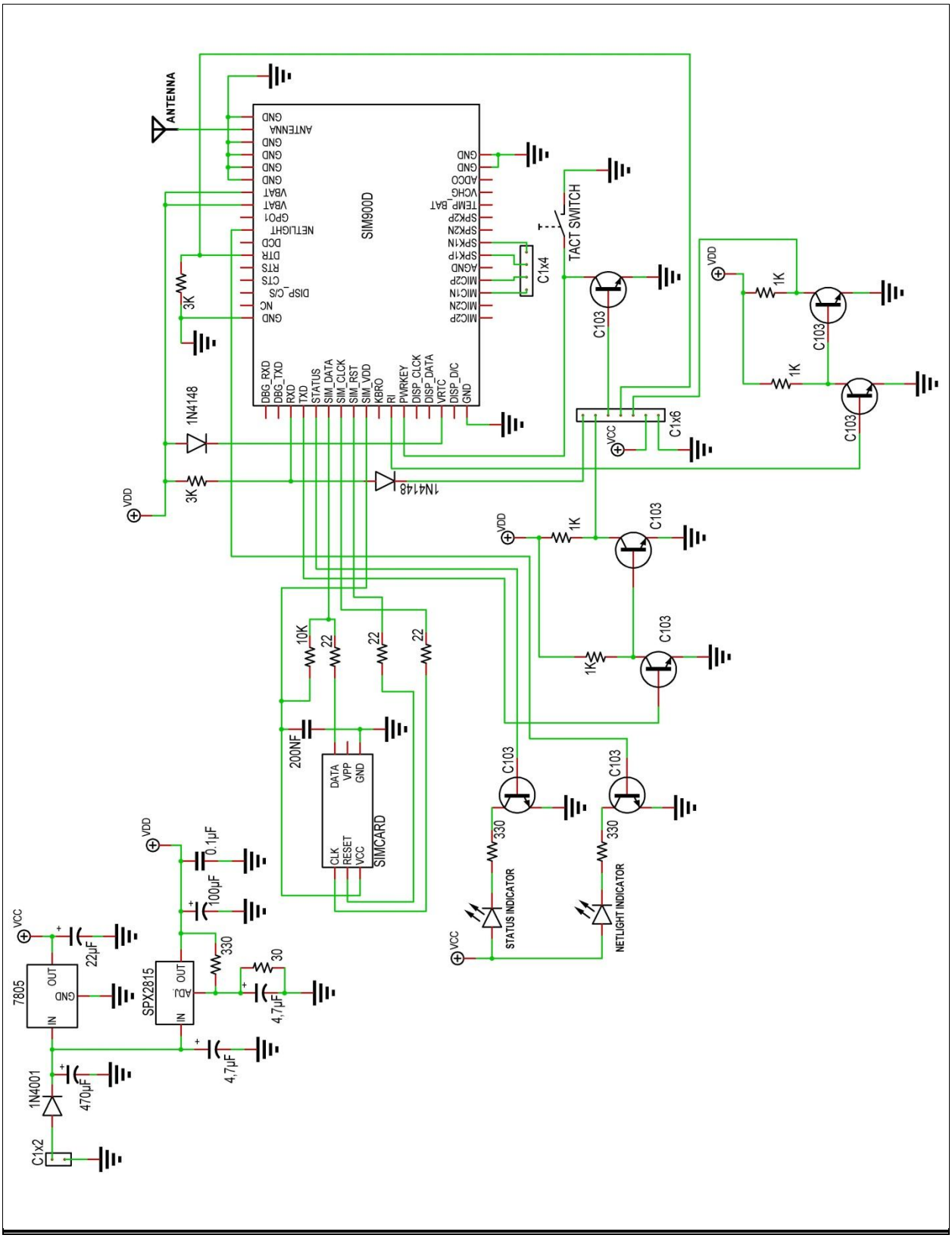


Figure 3.6: GSM Module Schematic Diagram

Software Development

The software is developed using Visual Studio 2010 where it has a built-in Visual Basic. The function of the software is to process the information received from the RFID Module. The program flow is as follows from Figure 3.7 wherein starts from receiving the unique code till it successfully send the data to parents and guardians.

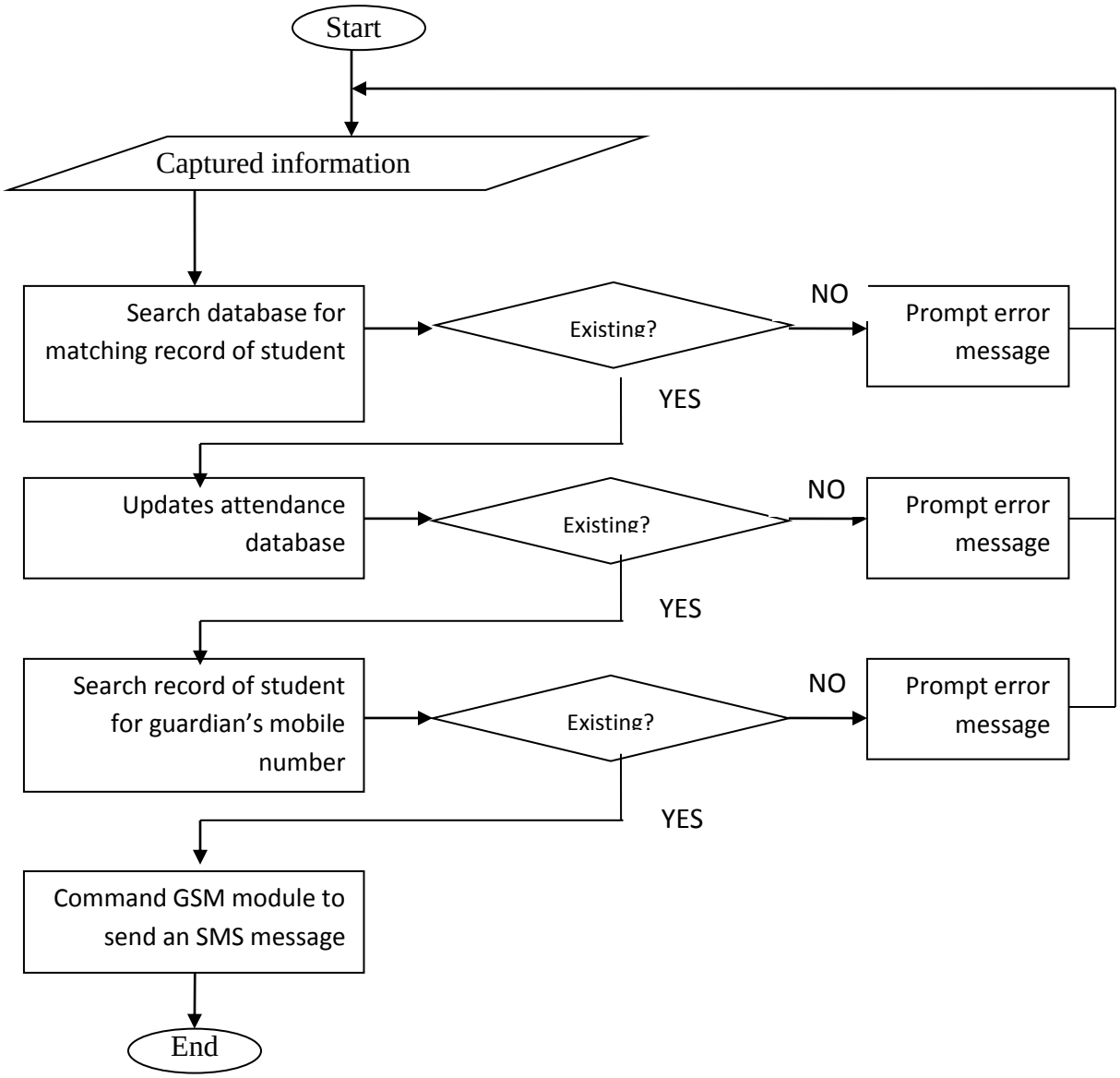


Figure 3.7: Program Flow Chart

The databases (with the following fields) will be:

Student record database:

Student Card Number Mobile number of parent/guardian Student Number Student Name Program Year

Attendance record database (Excel):

Student Number Name of student Time of log in Status

Chapter 4
Testing, Presentation, and Interpretation of Data

This chapter discusses the various tests conducted in relation with the stated objectives of the design to determine the functionality of the created prototype.

It discusses the tests conducted to determine the functionality of the created design and the capabilities of the whole system to find out if the prototype is working out well and to verify if the prototype had met the objectives of the design.

The proponent conducted tests to determine if the design is working and have followed the objectives of the design project. In performing the tests, the whole system was checked for percentage errors in the results. The program also checked for errors and bug related issues. In doing these tests, the proponent considered the objectives and the performance of the system design.

RFID card reader Test

Purpose: This activity aims to determine if the RFID card reader works with given several RFID tags and to check the capability of the Software Module in processing the encrypted data.

Assumption: The LED light on the prototype serves as the detector in which once the colour turns from GREEN to RED the RFID card reader is ready. Otherwise the RFID card reader does not work properly.

Procedure:

1) Prepare the software and hardware of the system.

2) Prepare the RFID that would be used for the identification of the user.

3) Turn on all power buttons for the test and configure the settings correctly.

4) When prompted to Attendance Form, log in by typing the given password first.

5) Start the testing by one student ID only then add more as the testing continuous.

6) Tap the RFID ID tagged into the reader and record whether the student was detected or not. Repeat the process for time intervals.

Note: Make sure to check the LED light, which turns green to red when ready.

RFID Card Reader Test		
No. of students All present	Expected Results (Display students Info)	Actual Results
1	YES	YES
2	YES	YES
3	YES	YES
4	YES	YES

5	YES	YES
6	YES	YES
7	YES	YES
8	YES	YES
9	YES	YES
10	YES	YES

Table 4.1: RFID Card Reader Test

Data Interpretation and Analysis

Table 4.1 shows that the RFID was able to display all information of each student ID that has been tapped despite of different number. This simply signifies that the RFID module is working properly. The Software module is also working properly because it was able to display the students' information.

SMS Sending Test

Purpose: This activity aimed to determine if the GSM module works after the tapping of the students who are absent in the RFID card reader and to check the capability of the Software Module in processing the encrypted data before actual automatic sending of SMS. It also shows the interval between SMS with one receiver only.

Assumption: Every SMS attempt, there should be a display showing that it has been sent. This is also the same message that the receiver will receive.

Procedure:

1) Prepare the software and hardware of the system.

2) Prepare the RFID that would be used for the identification of the user.

3) Turn on all power buttons for the test and configure the settings correctly.

4) When prompted to Attendance Form, log in by typing the given password first.

5) Start the testing by one student ID only then add more as the testing continuous.

6) Tap the RFID ID tagged into the reader and record whether the student was detected or not. Repeat the process for time intervals.

Note: Make sure to check the LED light, which turns red to green when ready.

7) Record the results after said testing.

SMS Sending Test			Interval	
No. of students Absent	Expected Results (Received SMS)	Actual Results	Expected Results 1-2 secs delay on each SMS receive	Actual Results
1	YES	YES	YES	YES
2	YES	YES	YES	YES
3	YES	YES	YES	YES
4	YES	YES	YES	YES
5	YES	YES	YES	YES
6	YES	YES	YES	YES
7	YES	YES	YES	YES
8	YES	YES	YES	YES
9	YES	YES	YES	YES
10	YES	YES	YES	YES

Table 4.2: SMS Sending Test

Data Interpretation and Analysis

Table 4.3 shows that Software module was working properly because it was able to process the encrypted data from the students ID. This also signifies that the GSM module was working properly because it was able to SEND SMS with the given interval between each text despite of different number of absent students.

Automated Attendance Test

Purpose: This activity aimed to determine if the design was able to solve the general problem which was the automation of attendance system.

Assumption: An excel file must be created, you are given the choice on which drive should be saved with desired filename.

Procedure:

1) Prepare the software and hardware of the system.

2) Prepare the RFID that would be used for the identification of the user.

3) Turn on all power buttons for the test and configure the settings correctly.

4) When prompted to Attendance Form, log in by typing the given password first.

5) Tap the ten RFID ID tagged into the RFID reader one at a time then which makes all students present. Followed by the next given condition.

Important: Make sure to check the LED light, which turns green to red when ready.

6) From the Student Attendance Dashboard, press File then the "Write Excel Report".

7) The software will prompt a question if the user wants to save the changes made on Book1 or the default record on Microsoft Excel Format. Press the "Save" button for it to save the latest report.

8) Open the Excel format to see if all the given condition is satisfied.

9) Record the results after said testing. Then proceed with the next condition.

Automated Attendance System		
List of Students	Expected Results (display info at Excel) "P"	Actual Results
1-Apple	YES	YES
2-Ben	YES	YES
3-Cathy	YES	YES
4-Donald	YES	YES
5-Ejay	YES	YES
6-Fred	YES	YES
7-Gerald	YES	YES
8-Harry	YES	YES
9-Irene	YES	YES
10-Unknown	NO	NO

Table 4.3: Automated Attendance System

Data Interpretation and Analysis

Table 4.4 shows that design was able to automate attendance system because it was able to record the attendance of present student in the classroom through Excel format. The Excel format had these values: First column must contained "Student Number", the second column "Name of the student", third column "Time of log in, last column showed the status of the student all present.

Another condition was no tapping occurred, this simply means that all students were supposed to be absent. It still displayed the said values but with a different student status which was "A" for absent.

Chapter 5

Conclusion and Recommendation

Conclusion

After performing different test and analysing different problems, the PC Interfaced Attendance System with SMS Capabilities achieved its objective to automate attendance in a typical classroom and informed parents or guardians if their child had logged in his respective classroom through SMS.

A reliable, secure, fast and an efficient system also had been developed replacing a manual and unreliable system. This system saved time, reduced the amount of work of the professor replaced the stationery material with electronic apparatus. Though a system with expected results had been developed, there is still some room for improvement.

Furthermore, the PC Interfaced Attendance System With SMS Capabilities gives an idea to other designers that are planning to incorporate SMS to their system.

Microsoft Visual Basic built-in Visual Studio have been used as the backend program of the system. It serves as the key for the storage of the students information, students log records or the incoming entry of students inside the classroom and the users account that access the software part of the system. Microsoft Access has been used for storing the main database of the program.

The message transmission and reception medium is the GSM modem. The GSM modem is the custom device of sending and receiving messages. It comprises of a GSM module, a SIM holder and RS232 converter.

The student monitoring system is set manually on the sending of SMS to the guardian once there is a student log session, both proper and improper. The logs will be triggered by the RFID tag while the system day and time for the absences. In case where the parents don't want to receive info texts, subscription and unsubscription are integrated within the system. There are other options such as message customization and just plain text messaging service. The system is also capable of saving weekly student log to a specific storage for future use by simply saving it through the Excel format.

Recommendation

After several tests have been performed by the researchers of the design, further studies are needed in order to improve the project.

First, in case this will be in mass produce, use the latest model of GSM module available in the market to be more proficient with the use of such equipment. While it will also be better if there will be a collaboration with the cellular network providers to decrease the expense for text messaging.

Secondly, enhance the incremental backup of the database by integrating third party software to the compress files in order to maximize the storage space. Besides using Microsoft Visual Studio, other better software can be used to increase and improve saving or storing capacity of its database.

Logs database should also be backup periodically for immediate retrieval upon request. Also this can be done to lessen the storage area of the database of the operational machine. Regards to RFID Module, other sensing devices can be used as substitute like instead of doing the regular way which is tapping, a student will just pass through and its attendance will automatically be recorded or identified by the monitoring device.

A biometrics system can also be an alternative, such as fingerprints and retina data, replacing the RFID tags for better security in checking the students attendance.

REFERENCES

Calupitan, et a (2006). MCU-Based Operated Text Booth.

Nokia Telecommunications (1999). *GSM System Training (SYSTRA)*, NTC CTXX 1985 en Issue 3.

Cunanan, et al(2006). *Mainframe Auto-Login using RFID*. Retrieved 2009, from http://news.cnet.com/8301-17938_105-10252222-1.html

Philips-nxp electronics (2001). *Mifare RFID Card Class Tag Information*.

Dong Ngo(July 2009). *Japanese university uses IPhone to keep tab of students*. Retrieved from http://news.cnet.com/8301-17938_105-10252222-1.html

"Mapua Cardinal Plus Description" Retrieved from http://www.mapua.edu.ph

Chris Savage, Parallax Tech Support (2008). *RFID Reader Software (Vb.net)*. Retrieved from, http://forums.parallax.com/forums/default.aspx?f=21&m=269675

Cellular Online (2000). *Hayes AT Command Sets*. Retrieved September 2, 2002, from http://www.cellular.co.za/hayesat.htm#Hayes%20AT%20Commands

ActiveXperts (1999). Network Monitoring Software, *ActiveXperts: SMS Messaging Server*. Retrieved 2004, from http://www.activeexperts.com

TAL Technologies (1999). Tech Support, TALtech web support: *USB to Serial Port Configuration.* Retrieved Jan 2007, from http://taltech.com/Tal_tech_web/suppport/usb2serial.html

Calasanz, Tristan H., (Jun 2000-Mar 2005). List of Projects Directed. Department of Electronics Communications and Computer Engineering, Ateneo de Manila University.

APPENDIX A

User's Manual

SAFETY GUIDELINES

The Automated Attendance design must be maintained regularly. Daily inspection must be maintained. Other components such as the CPU and peripherals must also be checked regularly if working properly. Failure to do so may result in less efficiency and failure.

The software component of the design (in Visual Basic#) must also be checked for errors and overflows in executing commands by the programmer. The database (in Microsoft Access) must be always secured as it contains confidential information. Database access, editing and deletion should only be done by system administrators.

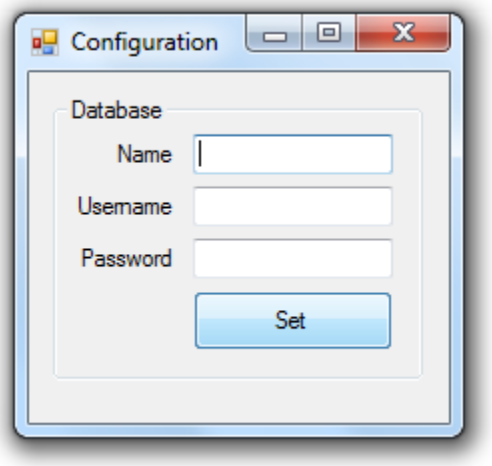
An overall physical inspection must be carried out at least after a day of operation. RFID Card Reader and GSM module must be carefully cleaned as it is a sensitive component. Look for damaged components, wire bends/stretches, and cracked casings and repair immediately.

OPERATING PROCEDURE

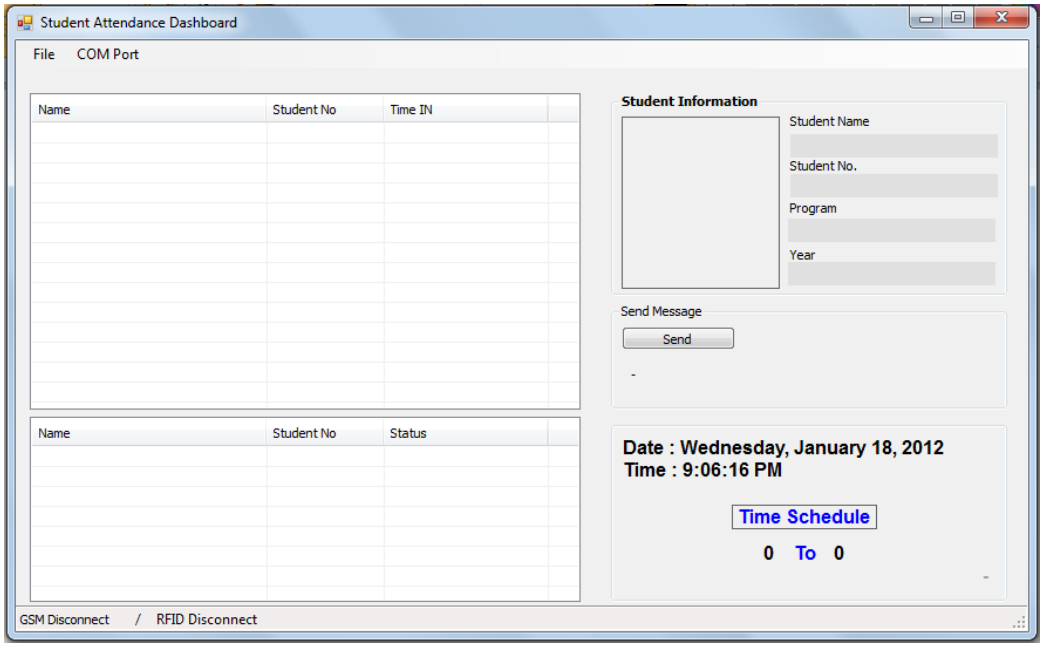
1. Properly connect the plug to a power source.
2. Both COM cable of the RFID and GSM module must be connected to the COM port of the CPU.

   Important: Make sure to identify which COM port is being used by each module once plug in to the CPU. This will be identified when selecting the correct port for each module afterwards.

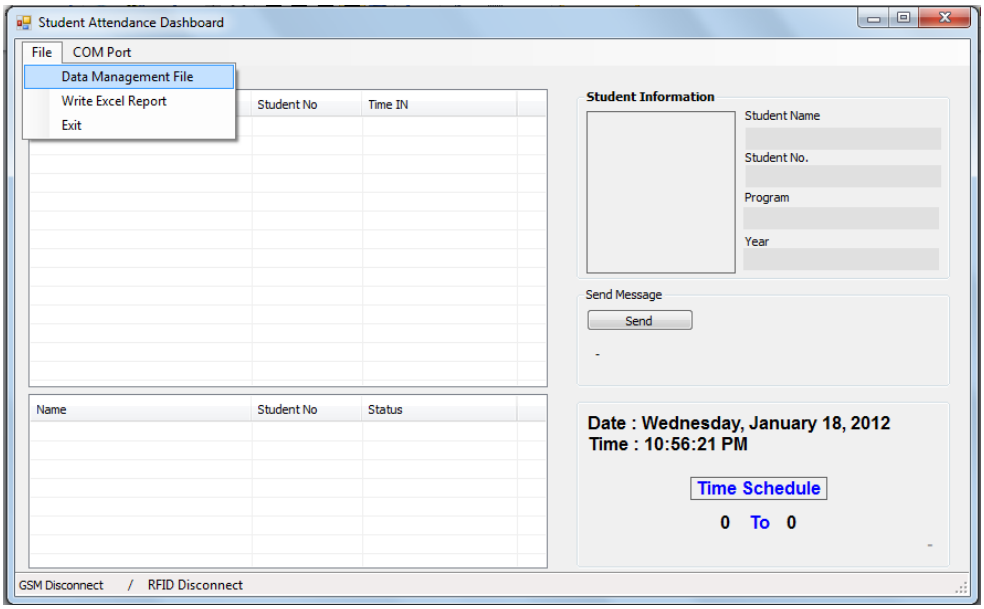
3. Run the Database Manager software and login to proceed. Type "admin" as the name, username and password.



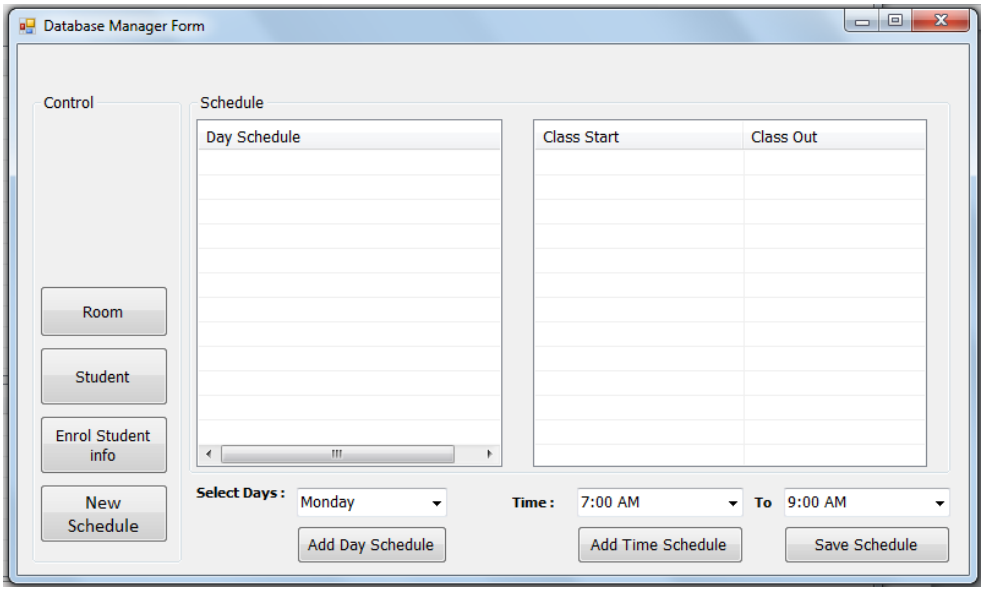
4. The Student Attendance Dashboard will appear wherein this is where checking of Attendance will begin. Profile of each student will appear to the screen once tapped. There is also a "Send" button where it is the Manual technique for sending. Actual Date and time is also present, plus room schedule for possible date and time.



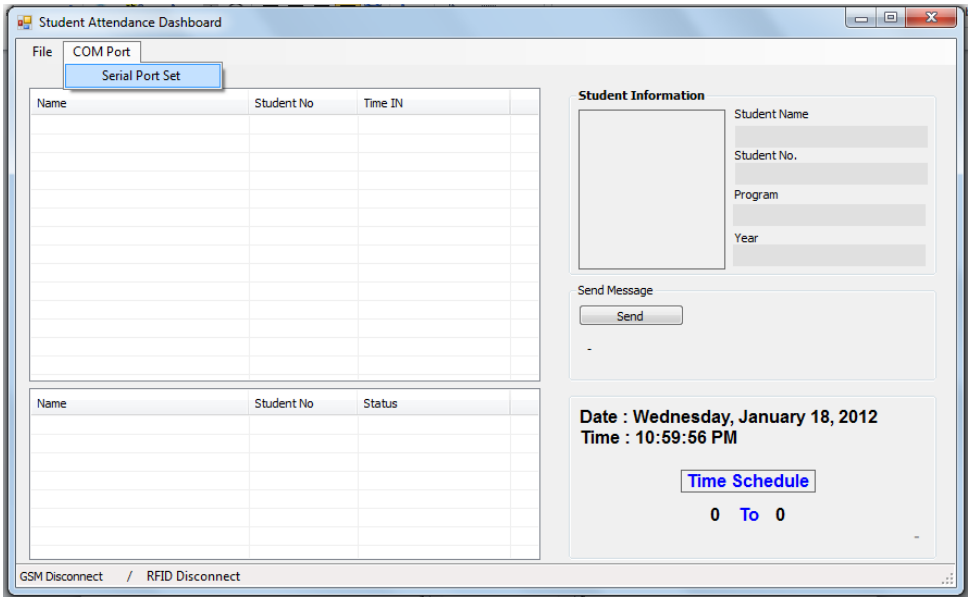
5. Press File then click ***Data Management File*** for creating a new class schedule. Provide all the necessary information or just select from the given choices above for the ***Room***, ***Student, Enrol Student info***, and New Schedule. Selecting New Schedule will simply delete the entire selected schedule like the selected days or the time when class start and ends.



Then:

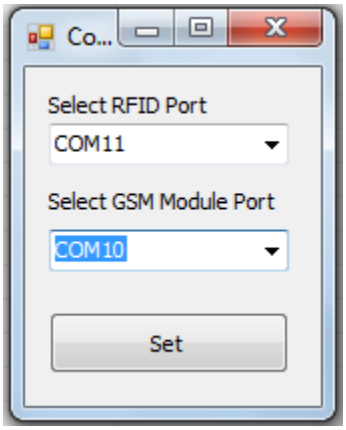


6. After creating a schedule, then can proceed on Selecting of Ports by pressing the ***COM Port*** menu.



7. Then ***Serial Port Set*** in selecting the right COM port for each module.

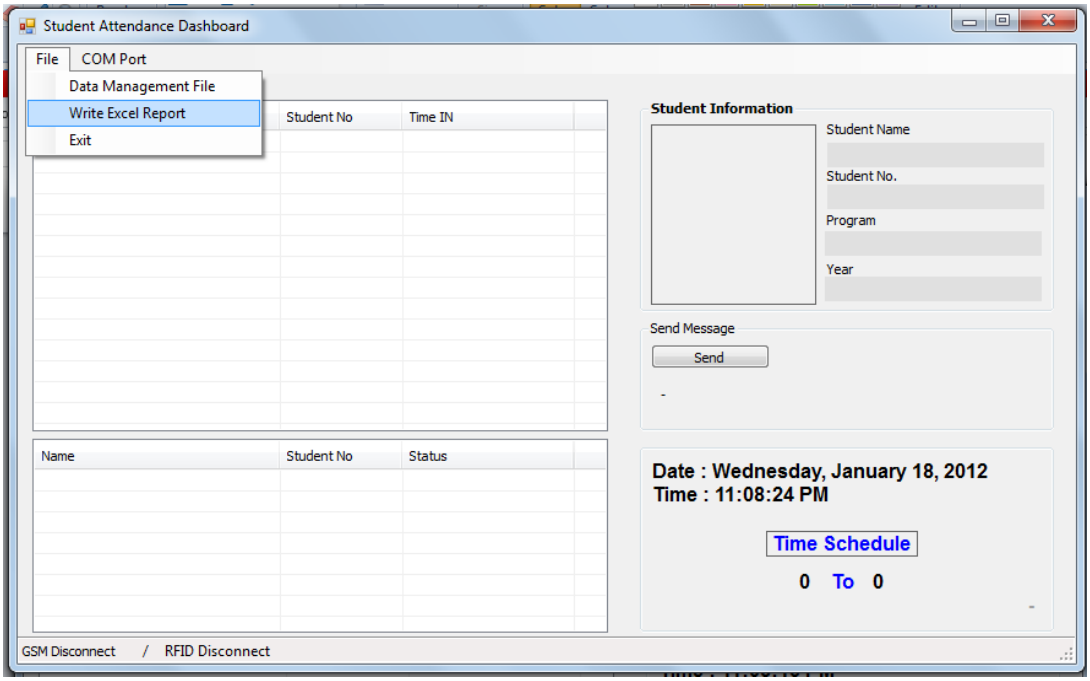
   Important: Make sure to identify correctly which COM port is being used by each module. Do not be confuse, might interchange two com ports that might results in less efficiency and failure.



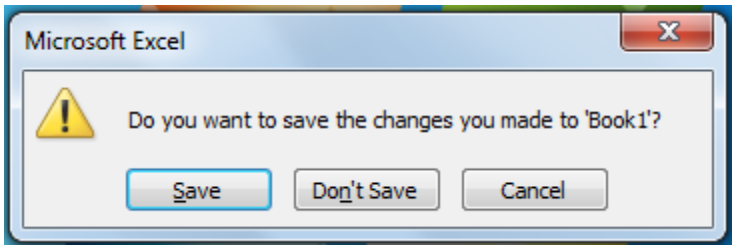
   Press the "Set" button afterwards, and checked for the status of each module just below the Student Attendance Dashboard. Whether it has been connected or disconnected. After that you can start tapping the RFID tagged to begin the checking of attendance depending if there is any available schedule on that day.

8. Next time when using the application, may directly proceed to Step 6 not passing Step 5 anymore if the entire schedule have been set already.
9. The sending of SMS is maybe done both automatically and manually. Automatic sending is implemented after certain given time, but if the users choose to do it manually. He may do so, because that's another alternative option of the user.

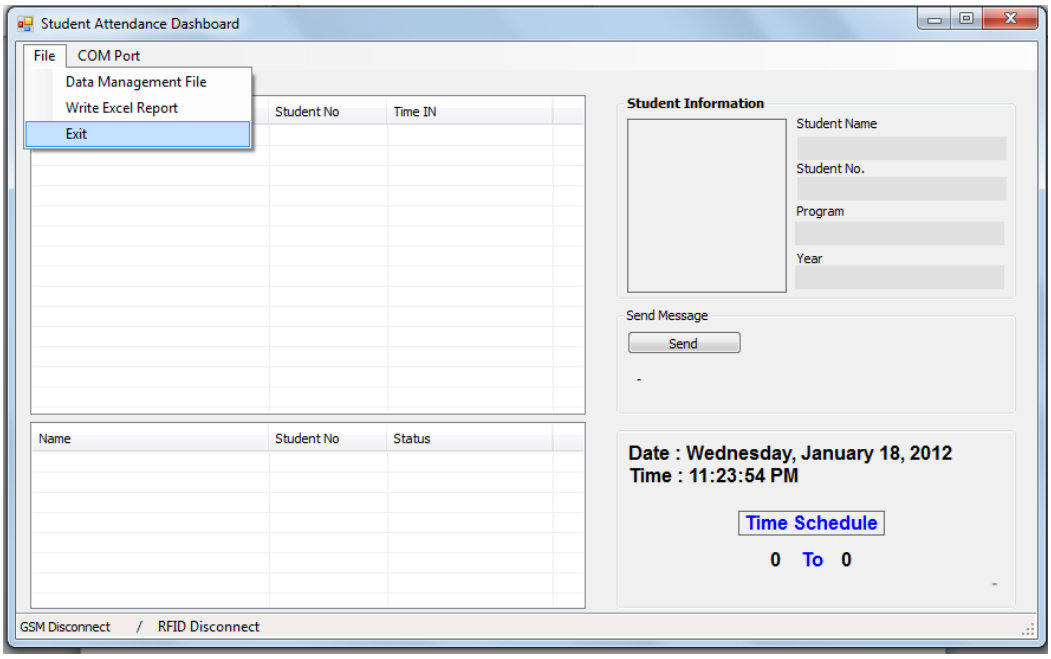
10. Last option will be the generating Attendance report through Microsoft Excel. From the Student Attendance Dashboard, press File then the "Write Excel Report".



11. After that software will prompt a question if the user wants to save the changes made on Book1 or the default record on Microsoft Excel Format. Press the "Save" button for it to save the latest report or "Don't Save" so that there will be no changes on the last report. "Cancel" button will simply quite everything not saving any record.



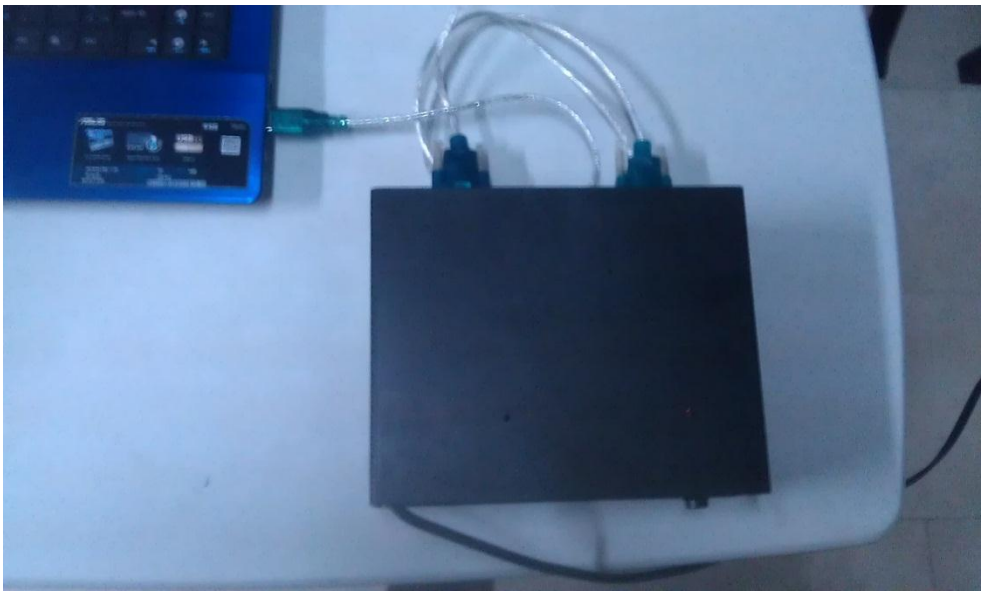
12. While to quit the Application, simply press the "Exit" selection just below the Write Excel Report. Once click it will automatically disappear. May also click the X button for quicker technique in quitting the application.



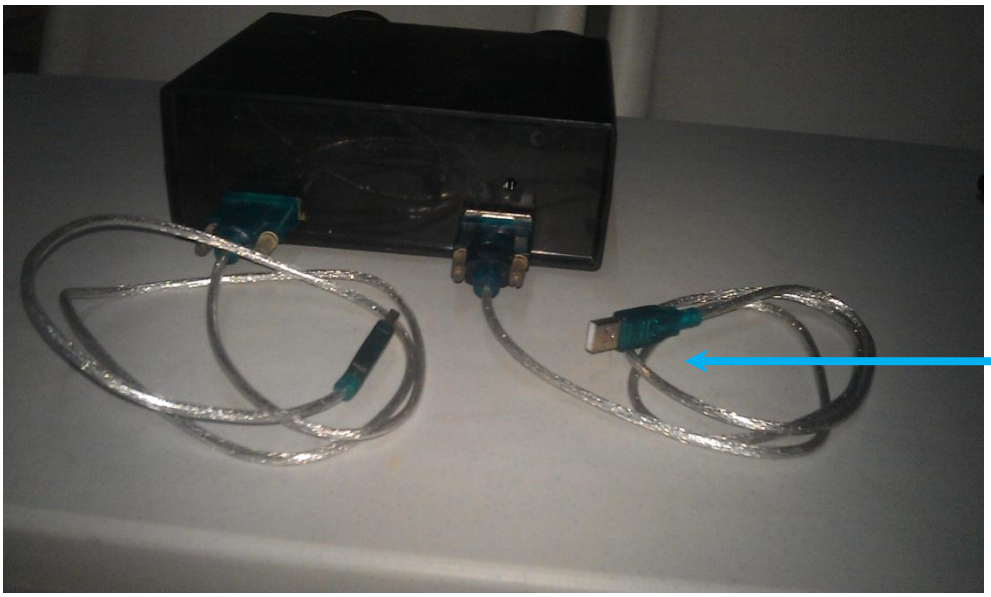
APPENDIX B

Prototype Pictures

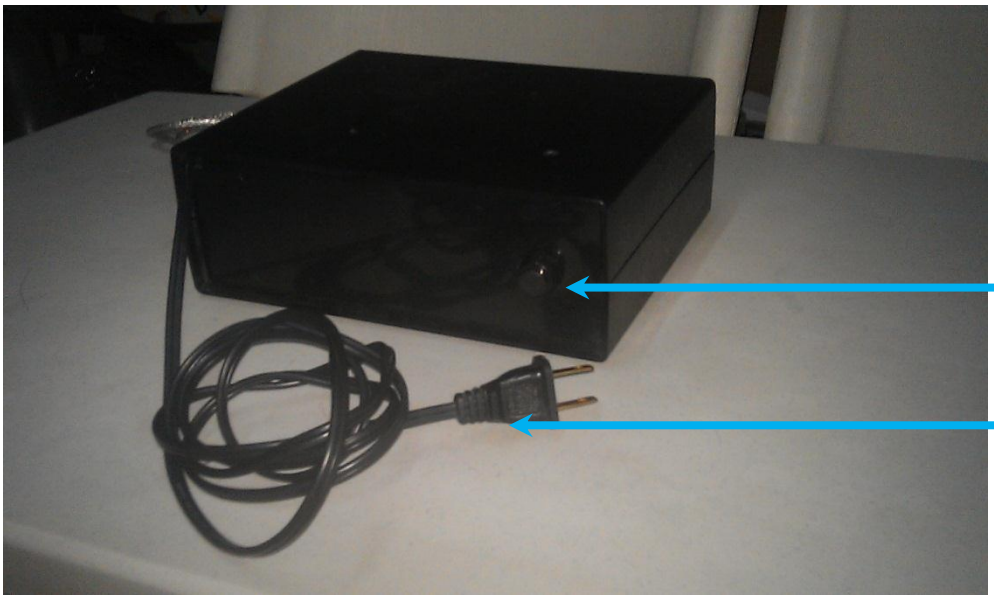
Actual Prototype

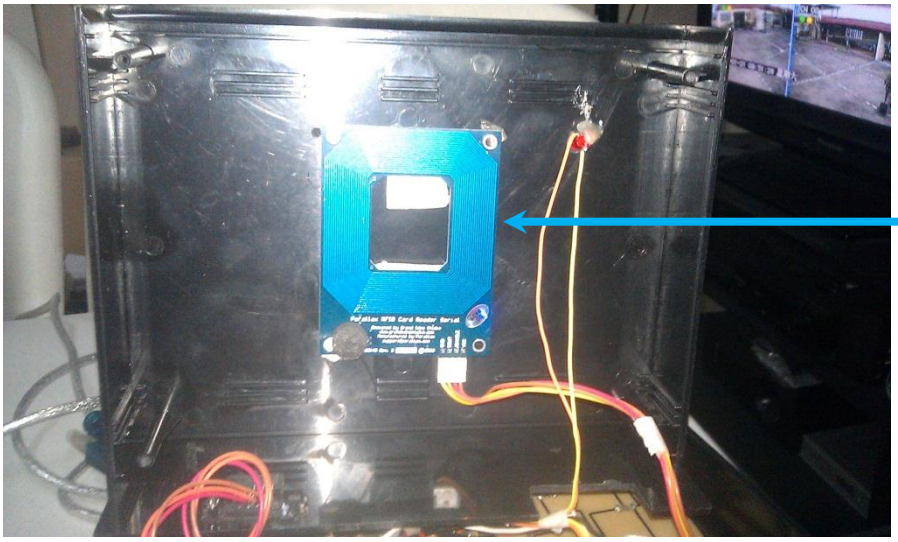


Prototype Top View

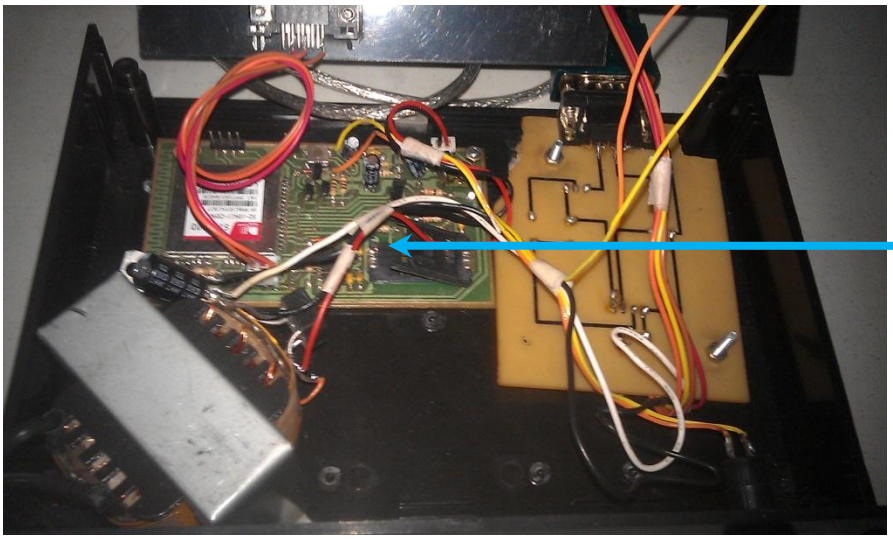


Prototype REAR VIEW





RFID Card Reader



SIM900D
GSM w/ GPRS modem

APPENDIX C

Program Listing

Database Manager Source Code:

```
Imports System
Imports System.Collections.Generic
Imports System.ComponentModel
Imports System.Data
Imports System.Drawing
Imports System.Text
Imports System.Windows.Forms
Imports System.IO.Ports


Public Class frmAttendance

    Dim RxDat As String
    Dim RxDat1 As String
    Dim rfID As String
    Dim cntLOG As Integer

    Dim sOld As String
    Dim sMobileNo As String
    Dim sSMSMsg As String = "Absent"
    Dim tmrClear As Integer
    Dim sendTemp As String
    Dim arrTime() As String

    Dim TimeEnbCounter As Integer = 0
    Dim sendEnbCounter As Integer = 0

    Private Sub Timer_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer.Tick

        On Error GoTo errh

        Dim i As Integer = 0
        Dim t As Integer = 0

        Dim tmp1, tmp2, tmp3, tmp4, tmp5, tmp6 As String

        lbltimeStart.Text = 0
        lbltimeEnd.Text = 0

        lblTimeDate.Text = "Date : " & Format(Now, "dddd, MMMM d, yyyy") & vbCr &
_
                           "Time : " & Format(Now, "h:mm:ss tt")

        If tmrClear = 15 Then
            tmrClear = 0
            lblName.Text = ""
            lblStudentNo.Text = ""
            lblProgram.Text = ""
            lblYear.Text = ""
        ElseIf tmrClear = 10 Then
            sOld = ""
            btnSendSMS.Enabled = True
        ElseIf tmrClear = 5 Then
            Call AbsentList()
            Call UpdateLog()
```

```
        End If

        tmrClear = tmrClear + 1

        Dim cn As New ADODB.Connection
        Dim cn1 As New ADODB.Connection
        Dim rs As New ADODB.Recordset
        Dim sql As String = "SELECT *FROM  q_TimePresent" ' WHERE LogDate=#" &
Format(Now, "MM/dd/yyyy") & "#"

        cn.Open(AccessConnect)
        rs.Open(sql, cn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)
        Dim sTimeSend As String = rs.Fields("RemTime").Value
        Label7.Text = sTimeSend

        lbltimeStart.Text = Format(rs.Fields("TimeStart").Value, "h:mm tt")
        lbltimeEnd.Text = Format(rs.Fields("TimeOut").Value, "h:mm tt")

        If Val(sTimeSend) = 20 Then TimeEnbCounter = TimeEnbCounter + 1
        If Val(sTimeSend) = 19 Then

            'TimeEnbCounter = 0
            TimeEnbCounter = 0
            sendEnbCounter = 0

        ElseIf Val(sTimeSend) = 5 Then

            cn1.Open(AccessConnect)
            cn1.Execute("DELETE FROM tbl_log")
            cn1.Close()

        End If

        If TimeEnbCounter = 1 Then

            Do Until rs.EOF = True
                tmp1 = rs.Fields("StudentNo").Value
                tmp2 = rs.Fields("StudentName").Value
                tmp3 = rs.Fields("MobileNo").Value
                tmp4 = rs.Fields("idStatus").Value
                tmp5 = "U"
                tmp6 = Format(Now, "MM/dd/yyyy h:mm:ss tt")

                If IsNumeric(tmp3) = True Then

                    cn.Execute("INSERT INTO tbl_timesend Values('" & tmp1 & "','"
& tmp2 & "','" & tmp3 & "','" & _
                                                                    tmp4 & "','"
& tmp5 & "','" & tmp6 & "')")
                End If

                rs.MoveNext()

            Loop

            sendEnbCounter = 1
            Timer2.Enabled = True
```

```
            EnbSending = True

        End If


        rs.Close()
        cn.Close()
        cn1.Close()
errh:

    End Sub

    Private Sub frmAttendance_Disposed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Disposed

        frmConfig.Show()

    End Sub

    Private Sub frmAttendance_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

        SPortRFID.Close()
        SPortRFID.DtrEnable = False

    End Sub

    Private Sub frmAttendance_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        Timer.Enabled = True
        Timer.Interval = 1000

        TimeEnbCounter = 0

        sendEnbCounter = 0

        Call UpdateLog()
        'AbsentList()

        StatusStrip1.Items.Item(0).Text = "GSM Disconnect   "
        StatusStrip1.Items.Item(1).Text = "  /  "
        StatusStrip1.Items.Item(2).Text = "RFID Disconnect"

        rfID = Nothing
        cntLOG = 0

    End Sub

    Private Sub UpdateLog()

        On Error Resume Next

        LVattendance.Clear()
        LVattendance.View = View.Details
        LVattendance.GridLines = True
        LVattendance.FullRowSelect = True
        LVattendance.Columns.Add("Name", 200, HorizontalAlignment.Left)
```

```
        LVattendance.Columns.Add("Student No", 100, HorizontalAlignment.Left)
        LVattendance.Columns.Add("Time IN", 140, HorizontalAlignment.Left)
        LVattendance.Columns.Add("Status", 140, HorizontalAlignment.Left)

        Dim cn As New ADODB.Connection
        Dim rs As New ADODB.Recordset
        'Dim sql As String = "SELECT *FROM q_scheduleTime WHERE chkLog='Yes'"
'"SELECT *FROM q_TimePresent WHERE idStatus='P' AND TimeStart <=#" & Format(Now,
"h:mm tt") & "# AND TimeOut >=#" & Format(Now, "h:mm tt") & "#"

        cn.Open(AccessConnect)
        rs.Open("SELECT *FROM q_scheduleTime WHERE chkLog='Yes'", cn,
ADODB.CursorTypeEnum.adOpenStatic, ADODB.LockTypeEnum.adLockPessimistic)
        rs.Fields.Refresh()

        Dim listItems As ListViewItem

        Do Until rs.EOF
            listItems = LVattendance.Items.Add(rs.Fields("StudentName").Value)
            listItems.SubItems.Add(rs.Fields("StudentNo").Value)
            listItems.SubItems.Add(rs.Fields("LoginNow").Value)
            rs.MoveNext()
        Loop

        rs.Close()
        cn.Close()

        rs = Nothing
        cn = Nothing

    End Sub

    Private Sub AbsentList()
        'On Error Resume Next

        LVAbsent.Clear()
        LVAbsent.View = View.Details
        LVAbsent.GridLines = True
        LVAbsent.FullRowSelect = True
        LVAbsent.Columns.Add("Name", 200, HorizontalAlignment.Left)
        LVAbsent.Columns.Add("Student No", 100, HorizontalAlignment.Left)
        LVAbsent.Columns.Add("Status", 140, HorizontalAlignment.Left)

        Dim cn As New ADODB.Connection
        Dim rs As New ADODB.Recordset
        Dim sql As String = "SELECT * FROM q_scheduleTime WHERE chkLog='No'"

        cn.Open(AccessConnect)
        rs.Open(sql, cn, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockPessimistic)
        rs.Fields.Refresh()

        Dim listItems As ListViewItem

        Do Until rs.EOF
            listItems = LVAbsent.Items.Add(rs.Fields("StudentName").Value)
            listItems.SubItems.Add(rs.Fields("StudentNo").Value)
            'listItems.SubItems.Add(rs.Fields("StudentName").Value)
```

```
            rs.MoveNext()
        Loop

        rs.Close()
        cn.Close()

    End Sub

    Private Sub SerialPortSetToolStripMenuItem1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SerialPortSetToolStripMenuItem1.Click

        If SPortRFID.IsOpen Then
            SPortRFID.Close()
        End If

        If SPortGSM.IsOpen Then
            SPortGSM.Close()
        End If

        'Timer2.Enabled = True
        frmPortConfig.Show()

    End Sub

    Public Delegate Sub myDelegate()

    Public Sub updateTextBox()

        TextBox1.AppendText(SPortGSM.ReadExisting)
        TextBox1.ScrollToCaret()

    End Sub

    Public Sub PortSettings()

        SPortGSM.Close()
        SPortGSM.PortName = frmPortConfig.CmbGSMPort.Text
        SPortGSM.BaudRate = 9600
        SPortGSM.Parity = IO.Ports.Parity.None
        SPortGSM.StopBits = IO.Ports.StopBits.One
        SPortGSM.DataBits = 8
        SPortGSM.Open()
        SPortGSM.Write("AT+CMGF=1" & vbCrLf)


        SPortRFID.Close()
        SPortRFID.PortName = frmPortConfig.CmbRFIDPort.Text
        SPortRFID.BaudRate = 2400
        SPortRFID.Parity = IO.Ports.Parity.None
        SPortRFID.StopBits = IO.Ports.StopBits.One
        SPortRFID.DataBits = 8
        SPortRFID.DtrEnable = True
        SPortRFID.Open()

    End Sub
```

```
    Private Sub SPortGSM_DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SPortGSM.DataReceived

        'TextBox1.BeginInvoke(New myDelegate(AddressOf updateTextBox), New
Object() {})
        RxDat = RxDat & SPortGSM.ReadExisting

    End Sub

    Private Sub SPortRFID_DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SPortRFID.DataReceived

        'System.Threading.Thread.Sleep(50)
        RxDat1 = RxDat1 & SPortRFID.ReadExisting

    End Sub

    Private Sub TimerGSM_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TimerGSM.Tick

        If SPortRFID.DtrEnable = True Then
            StatusStrip1.Items.Item(2).Text = "RFID connected"
        Else
            StatusStrip1.Items.Item(2).Text = "RFID Disconnect"
        End If

        If Len(RxDat1) = 12 Then

            If RxDat1 Like "*4800*" Then

                tmrClear = 0

                rfID = Mid(RxDat1, 2, Len(RxDat1) - 2)
                Label1.Text = rfID

                If sOld = rfID Then
                    RxDat = ""
                    RxDat1 = ""
                    Exit Sub
                Else
                    sOld = rfID
                    LogTimeIN(rfID)
                    Call UpdateLog()
                    Call AbsentList()
                End If
            Else
                TextBox1.Text = ""
            End If
        End If
        If RxDat Like "*OK*" Then
            TextBox1.Text = RxDat
            StatusStrip1.Items.Item(0).Text = "GSM connected"
            btnSendSMS.Enabled = True
        ElseIf RxDat Like "*>*" Then
            TextBox1.Text = RxDat
            SPortGSM.Write("Testing" & Chr(26))
        ElseIf RxDat Like "+CMGS*" Then
            btnSendSMS.Enabled = True
```

```
        End If
        RxDat = ""
        RxDat1 = ""

        If cntLOG = 15 Then
            cntLOG = 0
            rfID = Nothing
        End If


    End Sub

    Private Sub LogTimeIN(ByVal IDNo As String)

        On Error Resume Next

        Dim cn As New ADODB.Connection
        Dim rs As New ADODB.Recordset

        cn.Open(AccessConnect)
        rs.Open("SELECT *FROM tbl_student WHERE CardNo='" & IDNo & "'", cn,
ADODB.CursorTypeEnum.adOpenDynamic, ADODB.LockTypeEnum.adLockPessimistic)

        lblName.Text = ""
        lblName.Text = rs.Fields(2).Value
        lblStudentNo.Text = rs.Fields(1).Value
        lblProgram.Text = rs.Fields(4).Value
        lblYear.Text = rs.Fields(5).Value
        rs.Close()

        PictureMe.ImageLocation = App_Path() & "\Pictures\" & lblStudentNo.Text &
".jpg"

        If Len(lblName.Text) = 0 Then
            GoTo errh
        End If

        rs.Open("SELECT *FROM q_scheduleTime WHERE studentNo='" &
lblStudentNo.Text & "'", cn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)
        Dim stmpID As String = rs.Fields("StudentNo").Value
        rs.Close()

        If Len(stmpID) = 0 Then
            GoTo errh
        End If

        'Dim sqlcmd As String = "SELECT *FROM tbl_log WHERE StudentNo='" &
lblStudentNo.Text & "' " & _
        '                     "AND LogTime >=#" & Format(Now, "h:mm tt") & "# AND
LogDate=#" & Format(Now, "MM/dd/yyyy") & "#"

        Dim sqlcmd As String = "SELECT *FROM tbl_log WHERE StudentNo='" &
lblStudentNo.Text & "' " & _
                    "AND LogTime >=#" & Format(Now, "hh:mm:ss tt") & "# AND
LogDate=#" & Format(Now, "MM/dd/yyyy") & "#"
```

```
        rs.Open(sqlcmd, cn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)

        Dim stmp As String = rs.Fields("StudentNo").Value
        If Len(stmp) = 0 Then

            'cn.Execute("INSERT INTO tbl_log VALUES('" & lblStudentNo.Text & "','"
& _
            '                                          lblName.Text & "','" & _
            '                                          Format(Now, "h:mm tt") &
"','" & _
            '                                          Format(Now, "MM/dd/yyyy")
& "')")

            cn.Execute("INSERT INTO tbl_log VALUES('" & lblStudentNo.Text & "','"
& _
                                                      lblName.Text & "','" & _
                                                      Format(Now, "hh:mm:ss tt")
& "','" & _
                                                      Format(Now, "MM/dd/yyyy") &
"')")

        End If
errh:

        rs.Close()
        cn.Close()
        Call UpdateLog()

    End Sub

    Private Sub DataManagementFileToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DataManagementFileToolStripMenuItem.Click

        frmDMF.Show()

    End Sub

    Private Sub exitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles exitToolStripMenuItem.Click

        End

    End Sub


    Private Sub btnSendSMS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSendSMS.Click

        Dim sms As String = "AT+CMGS=""" & sMobileNo & """"
        SPortGSM.Write(sms & vbCr)
        System.Threading.Thread.Sleep(1000)
        SPortGSM.Write(sSMSMsg & Chr(26))
        btnSendSMS.Enabled = False

    End Sub
```

```
    Private Sub LVattendance_DoubleClick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles LVattendance.DoubleClick
        On Error Resume Next

        Dim i As Integer = LVattendance.SelectedItems.Item(0).Index
        Dim sIDNo As String = LVattendance.Items(i).SubItems(1).Text

        If SPortGSM.IsOpen Then
            btnSendSMS.Enabled = True
        End If

        sSMSMsg = "Present in Class"

        Dim cn As New ADODB.Connection
        Dim rs As New ADODB.Recordset

        cn.Open(AccessConnect)
        rs.Open("SELECT *FROM tbl_student WHERE StudentNo='" & sIDNo & "'", cn,
ADODB.CursorTypeEnum.adOpenDynamic, ADODB.LockTypeEnum.adLockPessimistic)

        sMobileNo = rs.Fields(3).Value
        lblMobileNo.Text = sMobileNo

        rs.Close()
        cn.Close()

    End Sub



    Private Sub LVAbsent_DoubleClick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles LVAbsent.DoubleClick
        On Error Resume Next

        Dim i As Integer = LVAbsent.SelectedItems.Item(0).Index
        Dim sIDNo As String = LVAbsent.Items(i).SubItems(1).Text

        If SPortGSM.IsOpen Then
            btnSendSMS.Enabled = True
        End If


        Dim cn As New ADODB.Connection
        Dim rs As New ADODB.Recordset

        sSMSMsg = "Absent in Class"

        cn.Open(AccessConnect)
        rs.Open("SELECT *FROM tbl_student WHERE StudentNo='" & sIDNo & "'", cn,
ADODB.CursorTypeEnum.adOpenDynamic, ADODB.LockTypeEnum.adLockPessimistic)

        sMobileNo = rs.Fields(3).Value
        lblMobileNo.Text = sMobileNo
errh:
        rs.Close()
        cn.Close()

    End Sub
```

```
    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick

        Timer2.Enabled = False

        If SPortGSM.IsOpen Then

            If sendEnbCounter = 1 Then

                Dim cn As New ADODB.Connection
                Dim rs As New ADODB.Recordset

                cn.Open(AccessConnect)
                rs.Open("SELECT *FROM tbl_timesend WHERE remark = 'A'", cn,
ADODB.CursorTypeEnum.adOpenDynamic, ADODB.LockTypeEnum.adLockPessimistic)

                Do Until rs.EOF

                    Dim sCp As String = rs.Fields("MobileNo").Value
                    Dim sNm As String = rs.Fields("IDName").Value
                    Dim sRm As String = rs.Fields("remark").Value

                    Dim SMSCp As String = "AT+CMGS=""" & sCp & """"
                    sSMSMsg = "Dear Sir/Madam : " & sNm & " / " & IIf(sRm = "P",
"Present", "Absent")

                    lblautomsg.Text = sSMSMsg
                    Me.Refresh()

                    System.Threading.Thread.Sleep(6000)

                    SPortGSM.Write(SMSCp & vbCr)
                    System.Threading.Thread.Sleep(1000)
                    SPortGSM.Write(sSMSMsg & Chr(26))

                    rs.MoveNext()

                Loop

                cn.Execute("DELETE FROM tbl_timesend")

                lblautomsg.Text = ""

                cn.Close()
                rs = Nothing
                cn = Nothing

            End If

        End If

    End Sub
```

```
    Private Sub WriteExelReportToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
WriteExelReportToolStripMenuItem.Click

        On Error Resume Next

        Dim cn As New ADODB.Connection
        Dim rs As New ADODB.Recordset
        Dim rs1 As New ADODB.Recordset
        Dim a, b, c As String

        Dim MyFolder As String = "E:\SMSNotification\"
        Dim oExcel As Object
        Dim oBook As Object
        Dim oSheet As Object

        'Start a new workbook in Excel.
        oExcel = CreateObject("Excel.Application")
        oBook = oExcel.Workbooks.Add

        'Add data to cells of the first worksheet in the new workbook.
        oSheet = oBook.Worksheets(1)
        oSheet.Range("A1").Value = "Student No"
        oSheet.Range("B1").Value = "Student Name"
        oSheet.Range("C1").Value = "Time IN"
        oSheet.Range("D1").Value = "Status"
        oSheet.Range("A1:B1:C1:D1").Font.Bold = True

        Dim XcellRow As Integer
        Dim XcellCollA As String
        Dim XcellCollB As String
        Dim XcellCollC As String
        Dim XcellCollD As String

        'cn.Open(AccessConnect)
        'rs.Open("SELECT *FROM tbl_log", cn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)

        XcellRow = 2

        cn.Open(AccessConnect)
        rs.Open("SELECT *FROM q_TimePresent", cn,
ADODB.CursorTypeEnum.adOpenDynamic, ADODB.LockTypeEnum.adLockPessimistic)

        Do Until rs.EOF

            XcellCollA = "A" & CStr(XcellRow)
            XcellCollB = "B" & CStr(XcellRow)
            XcellCollC = "C" & CStr(XcellRow)
            XcellCollD = "D" & CStr(XcellRow)

            oSheet.Range(XcellCollA).Value = rs.Fields("StudentNo").Value
            oSheet.Range(XcellCollB).Value = rs.Fields("StudentName").Value
            oSheet.Range(XcellCollC).Value = rs.Fields("LogTime").Value
            oSheet.Range(XcellCollD).Value = rs.Fields("idStatus").Value
```

```
            XcellRow = XcellRow + 1

            rs.MoveNext()

        Loop

        oSheet.Range("A:A").EntireColumn.AutoFit()
        oSheet.Range("B:B").EntireColumn.AutoFit()
        oSheet.Range("C:C").EntireColumn.AutoFit()
        oSheet.Range("D:D").EntireColumn.AutoFit()

        rs.Close()
        cn.Close()

        'Save the Workbook and quit Excel.
        oBook.SaveAs(MyFolder & "Book1.xls")
        oSheet = Nothing
        oBook = Nothing
        oExcel.Quit()
        oExcel = Nothing
        GC.Collect()

errh:

    End Sub


    Private Sub lbltimeStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles lbltimeStart.Click

    End Sub
End Class


    Private Sub lbltimeStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles lbltimeStart.Click

    End Sub
End Class
```

Public Class frmConfig

```
    Private Sub frmConfig_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load



    End Sub

    Private Sub btnSet_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSet.Click
        'On Error GoTo exth

        Dim cn As New ADODB.Connection
        Dim rs As New ADODB.Recordset

        cn.Open(AccessConnect)
```

```
        rs.Open("SELECT *FROM tbl_users WHERE UserName='" & txtUserName.Text & "'"
& _
                "AND Password='" & txtPassword.Text & "' AND LogName='" &
txtName.Text & "'", cn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)

        If Len(rs.Fields(0).Value) > 0 Then
            frmAttendance.Show()
            Me.Dispose(False)
        Else
            lblNote.Text = "invalid log"
        End If

        txtName.Text = ""
        txtUserName.Text = ""
        txtPassword.Text = ""

        rs.Close()
        cn.Close()
exth:

    End Sub

    Private Sub txtName_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtName.TextChanged

    End Sub
End Class
```

Public Class frmDMF

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        CmbDays.Text = "Monday"
        CmbDays.Items.Add("Monday")
        CmbDays.Items.Add("Tuesday")
        CmbDays.Items.Add("Wednesday")
        CmbDays.Items.Add("Thursday")
        CmbDays.Items.Add("Friday")
        CmbDays.Items.Add("Saturday")

        CmbTimeStart.Text = "7:00 AM"
        CmbTimeStart.Items.Add("7:00 AM")
        CmbTimeStart.Items.Add("7:30 AM")
        CmbTimeStart.Items.Add("8:30 AM")
        CmbTimeStart.Items.Add("9:00 AM")
        CmbTimeStart.Items.Add("9:30 AM")
        CmbTimeStart.Items.Add("10:00 AM")
        CmbTimeStart.Items.Add("10:30 AM")
        CmbTimeStart.Items.Add("11:00 AM")
        CmbTimeStart.Items.Add("11:30 AM")
        CmbTimeStart.Items.Add("12:00 AM")

        CmbTimeStart.Items.Add("1:00 PM")
        CmbTimeStart.Items.Add("1:30 PM")
        CmbTimeStart.Items.Add("2:00 PM")
```

```
        CmbTimeStart.Items.Add("2:30 PM")
        CmbTimeStart.Items.Add("3:00 PM")
        CmbTimeStart.Items.Add("3:30 PM")
        CmbTimeStart.Items.Add("4:00 PM")

        CmbTimeEnd.Text = "9:00 AM"

        CmbTimeEnd.Items.Add("9:00 AM")
        CmbTimeEnd.Items.Add("9:30 AM")
        CmbTimeEnd.Items.Add("10:00 AM")
        CmbTimeEnd.Items.Add("10:30 AM")
        CmbTimeEnd.Items.Add("11:00 AM")
        CmbTimeEnd.Items.Add("11:30 AM")
        CmbTimeEnd.Items.Add("12:00 AM")

        CmbTimeEnd.Items.Add("1:00 PM")
        CmbTimeEnd.Items.Add("1:30 PM")
        CmbTimeEnd.Items.Add("2:00 PM")
        CmbTimeEnd.Items.Add("2:30 PM")
        CmbTimeEnd.Items.Add("3:00 PM")
        CmbTimeEnd.Items.Add("3:30 PM")
        CmbTimeEnd.Items.Add("4:00 PM")
        CmbTimeEnd.Items.Add("4:30 PM")
        CmbTimeEnd.Items.Add("5:00 PM")
        CmbTimeEnd.Items.Add("5:30 PM")
        CmbTimeEnd.Items.Add("6:00 PM")
        CmbTimeEnd.Items.Add("6:30 PM")
        CmbTimeEnd.Items.Add("7:00 PM")

        btnSaveSchedules.Enabled = True


        LVDaySchedule.Clear()
        LVDaySchedule.View = View.Details
        LVDaySchedule.GridLines = True
        LVDaySchedule.FullRowSelect = True
        LVDaySchedule.Columns.Add("Day Schedule", 270, HorizontalAlignment.Left)

        LVTimeSchedule.Clear()
        LVTimeSchedule.View = View.Details
        LVTimeSchedule.GridLines = True
        LVTimeSchedule.FullRowSelect = True
        LVTimeSchedule.Columns.Add("Class Start", 170, HorizontalAlignment.Left)
        LVTimeSchedule.Columns.Add("Class Out", 170, HorizontalAlignment.Left)

        'Call loadRecord()

    End Sub

    Private Sub btnAddDaySch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddDaySch.Click

        LVDaySchedule.Items.Add(CmbDays.Text)

    End Sub

    Private Sub btnAddTimeSch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddTimeSch.Click
```

```
        'LVTimeSchedule.Items.Add(CmbTimeStart.Text & " - " & CmbTimeEnd.Text)
        Dim listItems As ListViewItem

        listItems = LVTimeSchedule.Items.Add(CmbTimeStart.Text)
        listItems.SubItems.Add(CmbTimeEnd.Text)

    End Sub

    Private Sub LVTimeSchedule_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles LVTimeSchedule.SelectedIndexChanged

        btnDelTime.Visible = True

    End Sub

    Private Sub btnDelTime_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDelTime.Click
        'On Error Resume Next
        Dim i As Integer = LVTimeSchedule.SelectedItems.Item(0).Index
        Dim sTime As String = LVTimeSchedule.Items(i).SubItems(0).Text
        'LVTimeSchedule.SelectedItems(0).Remove()
        LVTimeSchedule.Items.Item(i).Remove()

        Dim cn As New ADODB.Connection
        cn.Open(AccessConnect)
        cn.Execute("DELETE FROM tbl_Timeschedule WHERE TimeSchedule='" & sTime &
"'")
        cn.Close()

    End Sub

    Private Sub btnDayTime_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDayTime.Click
        On Error GoTo 0

        Dim i As Integer = LVDaySchedule.SelectedItems.Item(0).Index
        Dim sDay As String = LVDaySchedule.SelectedItems(0).Text

        'LVTimeSchedule.SelectedItems(0).Remove()
        LVDaySchedule.Items.Item(i).Remove()

        Dim cn As New ADODB.Connection
        cn.Open(AccessConnect)
        cn.Execute("DELETE FROM tbl_schedule WHERE Dayschedule='" & sDay & "'")
        cn.Execute("DELETE FROM tbl_Timeschedule WHERE DaySch='" & sDay & "'")
        cn.Close()

        'Call loadRecord()
        LVTimeSchedule.Items.Clear()
        btnDayTime.Visible = False

    End Sub

    Private Sub LVDaySchedule_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles LVDaySchedule.Click

        On Error GoTo errh
```

```
        btnDayTime.Visible = True
        Dim sDay As String = LVDaySchedule.SelectedItems(0).Text

        CmbDays.Text = sDay

        Dim cn As New ADODB.Connection
        Dim rs As New ADODB.Recordset

        sqlcmd = "SELECT *FROM tbl_Timeschedule WHERE DaySch='" & sDay & "'"
        cn.Open(AccessConnect)
        rs.Open(sqlcmd, cn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)
        Dim listItems1 As ListViewItem


        LVTimeSchedule.Items.Clear()

        Do Until rs.EOF
            listItems1 = LVTimeSchedule.Items.Add(rs.Fields(1).Value)
            rs.MoveNext()
        Loop

        rs.Close()
        cn.Close()
errh:
    End Sub

    Private Sub btnSaveSchedules_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSaveSchedules.Click

        Dim t As Integer
        Dim i As Integer = LVDaySchedule.Items.Count
        Dim l As Integer = LVTimeSchedule.Items.Count

        If i = 0 And l = 0 Then MsgBox("Schedule Required") : Exit Sub
        If Len(lblStudentNo.Text) < 2 And Len(lblRoom.Text) < 2 Then
            MsgBox("Student No. and Room No. Required")
            Exit Sub
        End If

        Dim sDaySch As String = LVDaySchedule.Items(t).SubItems(0).Text


        Dim Result As DialogResult
        Dim Buttons As MessageBoxButtons = MessageBoxButtons.YesNo
        Result = MessageBox.Show("Do you want to save", "Add  New Student
Schedule", Buttons)

        If Result = 6 Then

            Dim cn As New ADODB.Connection
            Dim rs As New ADODB.Recordset
            cn.Open(AccessConnect)

            For t = 0 To l - 1
                Dim strTime As String = LVTimeSchedule.Items(t).SubItems(0).Text
                Dim endTime As String = LVTimeSchedule.Items(t).SubItems(1).Text
```

```
                cn.Execute("INSERT INTO tbl_Schedule VALUES('" & lblStudentNo.Text
& "','" & sDaySch & "','" & _
                                                                    strTime & "','" &
endTime & "','" & lblRoom.Text & "')")

            Next

            cn.Close()

        End If

    End Sub

    Public Sub loadRecord(ByVal sRoom As String)

        Dim cn As New ADODB.Connection
        Dim rs As New ADODB.Recordset

        LVDaySchedule.Items.Clear()
        sqlcmd = "SELECT *FROM tbl_schedule WHERE RoomNo='" & sRoom & "'"
        cn.Open(AccessConnect)
        rs.Open(sqlcmd, cn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)
        Dim listItems As ListViewItem

        Do Until rs.EOF
            listItems = LVDaySchedule.Items.Add(rs.Fields(1).Value)
            rs.MoveNext()
        Loop

        rs.Close()
        cn.Close()


    End Sub



    Private Sub btnLoadRoom_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLoadRoom.Click

        frmRoom.Show()

    End Sub


    Private Sub btnStudent_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStudent.Click

        frmStudentList.Show()

    End Sub

    Private Sub btnEnroll_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEnroll.Click

        frmNewStudent1.Show()
```

```
    End Sub


    Private Sub btnAddNew_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddNew.Click

        LVDaySchedule.Items.Clear()
        LVTimeSchedule.Items.Clear()
        lblStudentNo.Text = ""
        lblRoom.Text = ""

    End Sub
End Class
```

Imports System.IO.Ports

```
Public Class frmPortConfig

    Private Sub btnSet_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSet.Click

        Call frmAttendance.PortSettings()
        Me.Close()

    End Sub

    Private Sub frmPortConfig_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        Dim ports As String() = SerialPort.GetPortNames()

        Dim port As String
        For Each port In ports
            CmbRFIDPort.Items.Add(port)
            CmbGSMPort.Items.Add(port)
        Next port

        CmbRFIDPort.SelectedIndex = 0
        CmbGSMPort.SelectedIndex = 0

    End Sub
End Class
```

Public Class frmRoom

```
    Private Sub frmRoom_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        cmbSetRoom.Text = "RM101"
        cmbSetRoom.Items.Add("RM101")
        cmbSetRoom.Items.Add("RM102")
        cmbSetRoom.Items.Add("RM103")
        cmbSetRoom.Items.Add("CL001")

    End Sub
```

```
    Private Sub btnSet_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSet.Click

        frmDMF.lblRoom.Text = cmbSetRoom.Text
        'Call frmDMF.loadRecord(cmbSetRoom.Text)

        Me.Close()

    End Sub
End Class
```

Public Class frmStudentList

```
    Private Sub frmStudentList_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load


        LVStudentList.Clear()
        LVStudentList.View = View.Details
        LVStudentList.GridLines = True
        LVStudentList.FullRowSelect = True
        LVStudentList.Columns.Add("Student No", 100, HorizontalAlignment.Left)
        LVStudentList.Columns.Add("Student Name", 170, HorizontalAlignment.Left)
        LVStudentList.Columns.Add("Program", 60, HorizontalAlignment.Left)
        LVStudentList.Columns.Add("School Year", 100, HorizontalAlignment.Left)

        Dim cn As New ADODB.Connection
        Dim rs As New ADODB.Recordset

        sqlcmd = "SELECT *FROM tbl_Student"
        cn.Open(AccessConnect)
        rs.Open(sqlcmd, cn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)
        Dim listItems As ListViewItem

        Do Until rs.EOF
            listItems = LVStudentList.Items.Add(rs.Fields(1).Value)
            listItems.SubItems.Add(rs.Fields(2).Value)
            listItems.SubItems.Add(rs.Fields(4).Value)
            listItems.SubItems.Add(rs.Fields(5).Value)
            rs.MoveNext()
        Loop

        rs.Close()
        cn.Close()

    End Sub

    Private Sub LVStudentList_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles LVStudentList.Click
        On Error GoTo errh

        Dim i As Integer = LVStudentList.SelectedItems.Item(0).Index
        Dim sIDNo As String = LVStudentList.Items(i).SubItems(0).Text

        frmDMF.lblStudentNo.Text = sIDNo

errh:
```

```
    End Sub


    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click

        Me.Close()

    End Sub


End Class
```

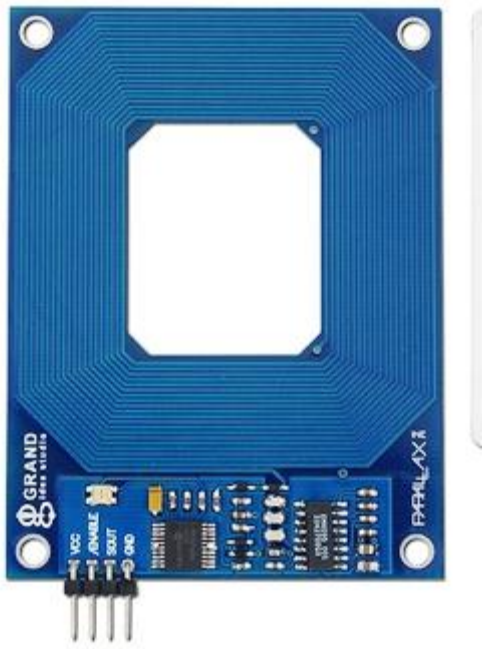
Module mdlMain

```
    Public RFIDport As Integer
    Public GSMport As Integer
    Public cn As ADODB.Connection
    Public cn1 As ADODB.Connection
    Public rs As ADODB.Recordset
    Public rs1 As ADODB.Recordset
    Public sqlcmd As String

    Public EnbSending As Boolean = False
    'Public AccessConnect = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
App_Path() & "student.mdb;Jet OLEDB:Database Password=12345;"
    'Public AccessConnect = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=E:\Visual Studio 2008\Projects\SMSNotification\student.mdb;User
Id=admin;Password=;"

    Public AccessConnect = "Driver={Microsoft Access Driver (*.mdb,
*.accdb)};Dbq=" & App_Path() & "student.mdb" & ";Uid=;Pwd=12345;"
    'Public AccessConnect = "Driver={Microsoft Access Driver (*.mdb*.mdb,
*.accdb)};" & _
    '                    "Dbq=student.mdb;" & _
    '                    "DefaultDir=" & App_Path() & ";" & _
    '                    "Uid=;Pwd=;"


    Public Sub main()

        '

    End Sub

    Public Function App_Path() As String

        'Return System.AppDomain.CurrentDomain.BaseDirectory()
        App_Path = System.IO.Path.GetFullPath("\- SMSNotification\")

    End Function

    Public Sub OpenRecordset()


    End Sub
```

```
End Module
```

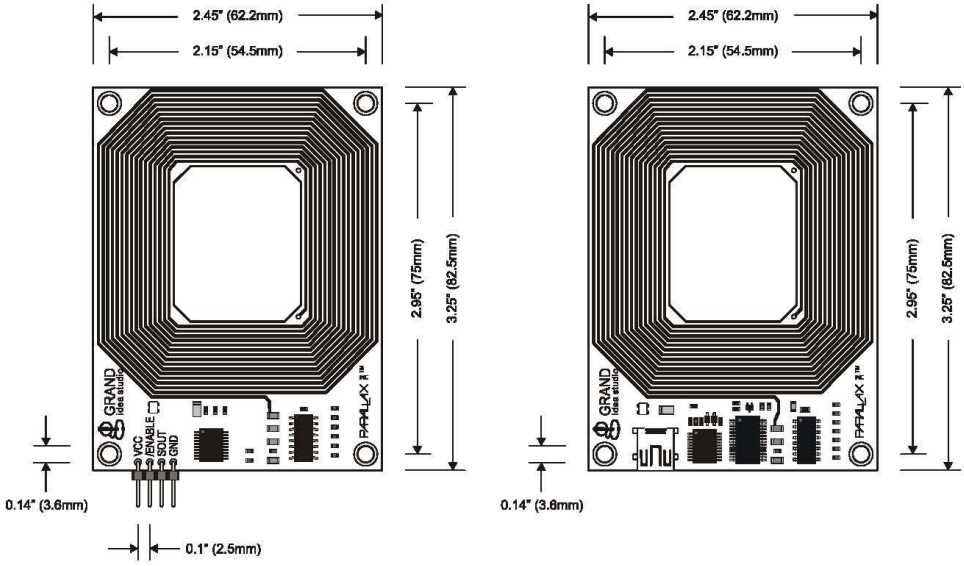
APPENDIX D

Data Sheets



RFID Card Reader Serial Radio freq

RFID Card Reader, Serial (#28140)
RFID Card Reader, USB (#28340)



Introduction

Designed in cooperation with Grand Idea Studio (www.grandideastudio.com), the Parallax Radio Frequency Identification (RFID) Card Readers provide a low-cost solution to read passive RFID transponder tags up to 4 inches away. The RFID Card Readers can be used in a wide variety of hobbyist and commercial applications, including access control, user identification, robotics navigation, inventory tracking, payment systems, car immobilization, and manufacturing automation. The RFID Card Reader is available in two versions: A TTL-level serial interface for use with a microcontroller and a USB interface for direct connection to a computer.

Features

- Low-cost method for reading passive, 125 kHz RFID transponder tags
- Two easy-to-use versions: Serial interface for microcontrollers and USB for direct connection to PC, Macintosh, or Linux machines
- Bi-color LED for visual indication of status

RFID Compatibility

The Parallax RFID Card Reader works exclusively with the EM Microelectronics EM4100-family of passive read-only transponder tags. Each transponder tag contains a unique, read-only identifier (one of 2^{40}, or 1,099,511,627,776 possible combinations).

A variety of different tag types and styles exist with the most popular made available from Parallax.

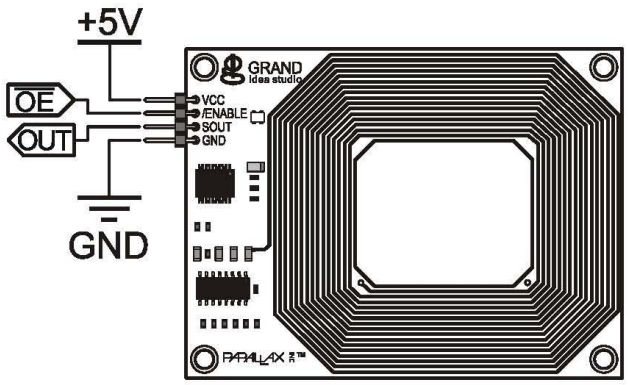
Connections (Serial)

The Parallax RFID Card Reader Serial version easily interfaces to any host microcontroller using only four connections (VCC, /ENABLE, SOUT, GND).

Pin	Pin Name	Type	Function
1	VCC	P	System power. +5V DC input.
2	/ENABLE	I	Module enable pin. Active LOW digital input. Bring this pin LOW to enable the RFID reader and activate the antenna.
3	SOUT	O	Serial output to host. TTL-level interface, 2400 bps, 8 data bits, no parity, 1 stop bit.
4	GND	G	System ground. Connect to power supply's ground (GND) terminal.

Note: Type: I = Input, O = Output, P = Power, G = Ground

Use the following example circuit for connecting the Parallax RFID Card Reader:



Connections (USB)

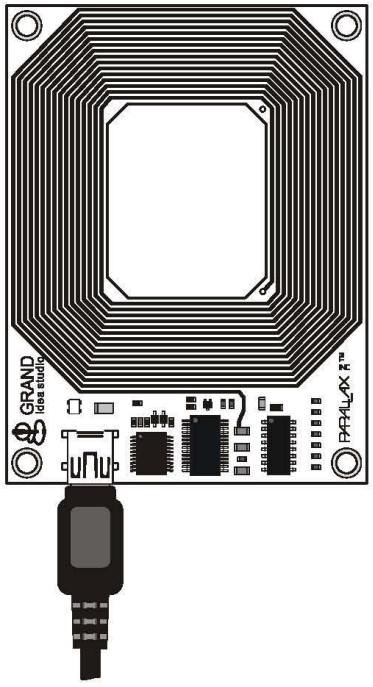
The Parallax RFID Card Reader USB version can be connected directly to any PC, Macintosh, or Linux machine that has a USB port and the appropriate drivers installed. The module is powered from the host computer's USB port and uses an industry-standard FTDI FT232R device to provide the USB connectivity. FTDI drivers are available from www.ftdichip.com/Drivers/VCP.htm.

Signal	Port Name	Function
RX	Serial Receive	Serial output to host. 2400 bps, 8 data bits, no parity, 1 stop bit.
DTR	Data Terminal Ready	Module enable. Bring the DTR line HIGH to enable the RFID reader and activate the antenna. Bring the DTR line LOW to disable the RFID reader.

When the Parallax RFID Card Reader is connected to the host computer, it will appear as a Virtual COM port and will have a COM port number automatically assigned to it. This COM port can be accessed by any software application, programming language, or interface that provides COM port connectivity and will allow you to read the data stream transmitted by the module.

An example program, including Visual Basic/VB.net source code, for reading tags in Windows XP/Vista is available on the Parallax website at http://forums.parallax.com/forums/default.aspx?f=21&m=269675.

The DEBUG window within the Parallax BASIC Stamp Editor (www.parallax.com/tabid/441/Default.aspx) provides functionality to set the state of a COM port's DTR line. Checking the DTR box in the toolbar will activate the RFID Card Reader.



Usage

A visual indication of the state of the RFID Card Reader is given with the on-board LED. When the module is successfully powered-up and is in an idle state, the LED will be GREEN. When the module is in an active state searching for or communicating with a valid tag, the LED will be RED.

The RFID Card Reader Serial version is activated via the /ENABLE pin on the module's 4-pin header. When the RFID Card Reader is powered and /ENABLE is pulled LOW, the module will enter the active state. When /ENABLE is pulled HIGH or left unconnected, the module will enter the idle state.

The RFID Card Reader USB version is activated via the DTR line of the USB Virtual COM port. When the DTR line is set HIGH, the module will enter the active state. When the DTR line is set LOW, the module will enter the idle state.

The face of the RFID tag should be held parallel to the front or back face of the antenna (where the majority of RF energy is emitted). If the tag is held sideways (for example, perpendicular to the antenna), you'll either get no reading or a poor reading distance. Only one transponder tag should be held up to the antenna at any time. The use of multiple tags at one time will cause tag collisions and the reader may not detect any of them. The tags available in the Parallax store have a read distance of approximately 4 inches. Actual distance may vary slightly depending on the size of the transponder tag and environmental conditions of the application.

Communication Protocol

All communication is 8 data bits, no parity, 1 stop bit, and least significant bit first (8N1) at 2400 bps.

The RFID Card Reader Serial version transmits data as 5V TTL-level, non-inverted asynchronous serial.

The RFID Card Reader USB version transmits the data through the USB Virtual COM Port driver. This allows easy access to the serial data stream from any software application, programming language, or interface that can communicate with a COM port.

When the RFID Card Reader is active and a valid RFID transponder tag is placed within range of the activated reader, the tag's unique ID will be transmitted as a 12-byte printable ASCII string serially to the host in the following format:

Start Byte (0x0A)	Unique ID Digit 1	Unique ID Digit 2	Unique ID Digit 3	Unique ID Digit 4	Unique ID Digit 5	Unique ID Digit 6	Unique ID Digit 7	Unique ID Digit 8	Unique ID Digit 9	Unique ID Digit 10	Stop Byte (0x0D)

The start byte and stop byte are used to easily identify that a correct string has been received from the reader (they correspond to line feed and carriage return characters, respectively). The middle ten bytes are the actual tag's unique ID. For example, for a tag with a valid ID of 0F0184F07A, the following bytes would be sent: 0x0A, 0x30, 0x46, 0x30, 0x31, 0x38, 0x34, 0x46, 0x30, 0x37, 0x41, 0x0D.

Interference

The Parallax RFID Card Reader, like many RF devices, may experience RF noise in its frequency range. This may cause the reader to transmit a spurious tag response when no tag is near the unit. This will not affect most uses of the RFID Card Reader. To avoid treating spurious responses as legitimate tags, it is recommended to read two responses in a row within a given amount of time (for example, one second) to ensure that you are reading a valid tag and not a "tag" generated by noise.

DC Characteristics

At V_{CC} = +5.0V and T_A = 25ºC unless otherwise noted

Parameter	Symbol	Test Conditions	Specification			Unit
			Min.	Typ.	Max.	
Supply Voltage	V_{CC}	---	4.5	5.0	5.5	V
Supply Current, Idle	I_{IDLE}	---	---	10	---	mA
Supply Current, Active	I_{CC}	---	---	100	200	mA
Input LOW voltage	V_{IL}	+4.5V <= V_{CC} <= +5.5V	---	---	0.8	V
Input HIGH voltage	V_{IH}	+4.5V <= V_{CC} <= +5.5V	2.0	---	---	V
Output LOW voltage	V_{OL}	V_{CC} = +4.5V	---	---	0.6	V
Output HIGH voltage	V_{OH}	V_{CC} = +4.5V	V_{CC} - 0.7	---	---	V

Absolute Maximum Ratings

Condition	Value
Operating Temperature	-40ºC to +85ºC
Storage Temperature	-55ºC to +125ºC
Supply Voltage (V_{cc})	+4.5V to +5.5V
Ground Voltage (V_{ss})	0V
Voltage on any pin with respect to V_{ss}	-0.3V to +7.0V

NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

RFID Technology Overview

Material in this section is based on information provided by the RFID Journal (www.rfidjournal.com).

Radio Frequency Identification (RFID) is a generic term for non-contacting technologies that use radio waves to automatically identify people or objects. There are several methods of identification, but the most common is to store a unique serial number that identifies a person or object on a microchip that is attached to an antenna. The combined antenna and microchip are called an "RFID transponder" or "RFID tag" and work in combination with an "RFID reader" (sometimes called an "RFID interrogator").

An RFID system consists of a reader and one or more tags. The reader's antenna is used to transmit radio frequency (RF) energy. Depending on the tag type, the energy is "harvested" by the tag's antenna and used to power up the internal circuitry of the tag. The tag will then modulate the electromagnetic waves generated by the reader in order to transmit its data back to the reader. The reader receives the modulated waves and converts them into digital data.

There are two major types of tag technologies. "Passive tags" are tags that do not contain their own power source or transmitter. When radio waves from the reader reach the chip's antenna, the energy is converted by the antenna into electricity that can power up the microchip in the tag (typically via inductive coupling). The tag is then able to send back any information stored on the tag by modulating the reader's electromagnetic waves. "Active tags" have their own power source and transmitter. The power source, usually a battery, is used to run the microchip's circuitry and to broadcast a signal to a reader. Due to the fact that passive tags do not have their own transmitter and must reflect their signal

to the reader, the reading distance is much shorter than with active tags. However, active tags are typically larger, more expensive, and require occasional service.

Frequency refers to the size of the radio waves used to communicate between the RFID system components. Just as you tune your radio to different frequencies in order to hear different radio stations, RFID tags and readers must be tuned to the same frequency in order to communicate effectively. RFID systems typically use one of the following frequency ranges: low frequency (or LF, around 125 kHz), high frequency (or HF, around 13.56 MHz), ultra-high frequency (or UHF, around 868 and 928 MHz), or microwave (around 2.45 and 5.8 GHz).

The read range of a tag ultimately depends on many factors: the frequency of RFID system operation, the power of the reader, and interference from other RF devices. Balancing a number of engineering trade-offs (antenna size v. reading distance v. power v. manufacturing cost), the Parallax RFID Card Reader's antenna was designed specifically for use with low-frequency (125 kHz) passive tags with a read distance of around 4 inches.

EM4102 RFID Card

EM MICROELECTRONIC - MARIN SA

A COMPANY OF THE SWATCH GROUP

EM4102

Read Only Contactless Identification Device

Description

The EM4102 (previously named H4102) is a CMOS integrated circuit for use in electronic Read Only RF Transponders. The circuit is powered by an external coil placed in an electromagnetic field, and gets its master clock from the same field via one of the coil terminals. By turning on and off the modulation current, the chip will send back the 64 bits of information contained in a factor programmed memory array.

The programming of the chip is performed by laser fusing of polysilicon links in order to store a unique code on each chip.

The EM4102 has several metal options which are used to define the code type and data rate. Data rates of 64, 32 and 16 periods of carrier frequency per data bit are available. Data can be coded as Manchester, Biphase or PSK.

Due to low power consumption of the logic core, no supply buffer capacitor is required. Only an external coil is needed to obtain the chip function. A parallel resonance capacitor of 78 pF is also integrated.

Features

- 64 bit memory array laser programmable
- Several options of data rate and coding available
- On chip resonance capacitor
- On chip supply buffer capacitor
- On chip voltage limiter
- Full wave rectifier on chip
- Large modulation depth due to a low impedance modulation device
- Operating frequency 100 - 150 kHz
- Very small chip size convenient for implantation
- Very low power consumption

Applications

- Animal implantable transponder
- Animal ear tag
- Industrial transponder

Typical Operating Configuration

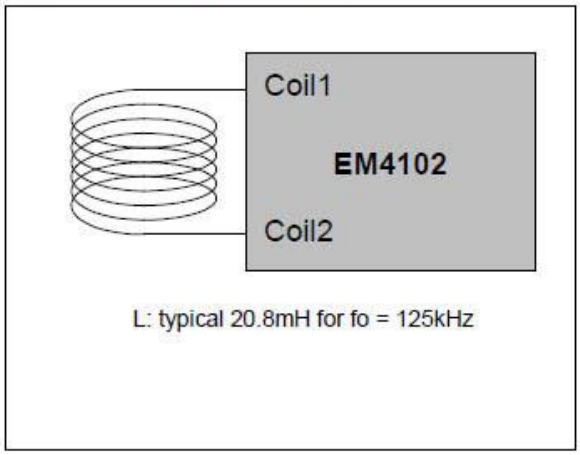


Fig. 1

Pin Assignment

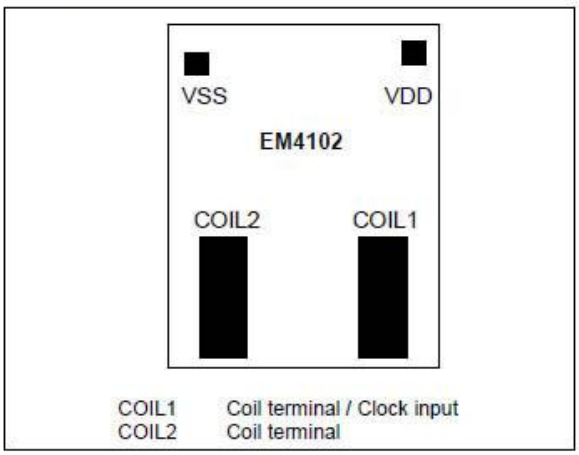


Fig. 2

Absolute Maximum Ratings

Parameter	Symbol	Conditions
Maximum DC Current forced on COIL1 & COIL2	I_{COIL}	±30mA
Power Supply	V_{DD}	-0.3 to 7.5V
Storage Temp. Die form Storage Temp. PCB form	T_{store} T_{store}	-55 to +200°C -55 to +125°C
Electrostatic discharge maximum to MIL-STD-883C method 3015	V_{ESD}	2000V

Stresses above these listed maximum ratings may cause permanent damages to the device. Exposure beyond specified operating conditions may affect device reliability or cause malfunction.

Handling Procedures

This device has built-in protection against high static voltages or electric fields; however, anti-static precautions must be taken as for any other CMOS component. Unless otherwise specified, proper operation can only occur when all terminal voltages are kept within the voltage range. Unused inputs must always be tied to a defined logic voltage level.

Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Operating Temp.	T_{op}	-40		+85	°C
Maximum Coil Current	I_{COIL}	-10		10	mA
AC Voltage on Coil	V_{coil}	3	14*		Vpp
Supply Frequency	f_{coil}	100		150	kHz

*) The AC Voltage on Coil is limited by the on chip voltage limitation circuitry. This is according to the parameter I_{coil} in the absolute maximum ratings.

System Principle

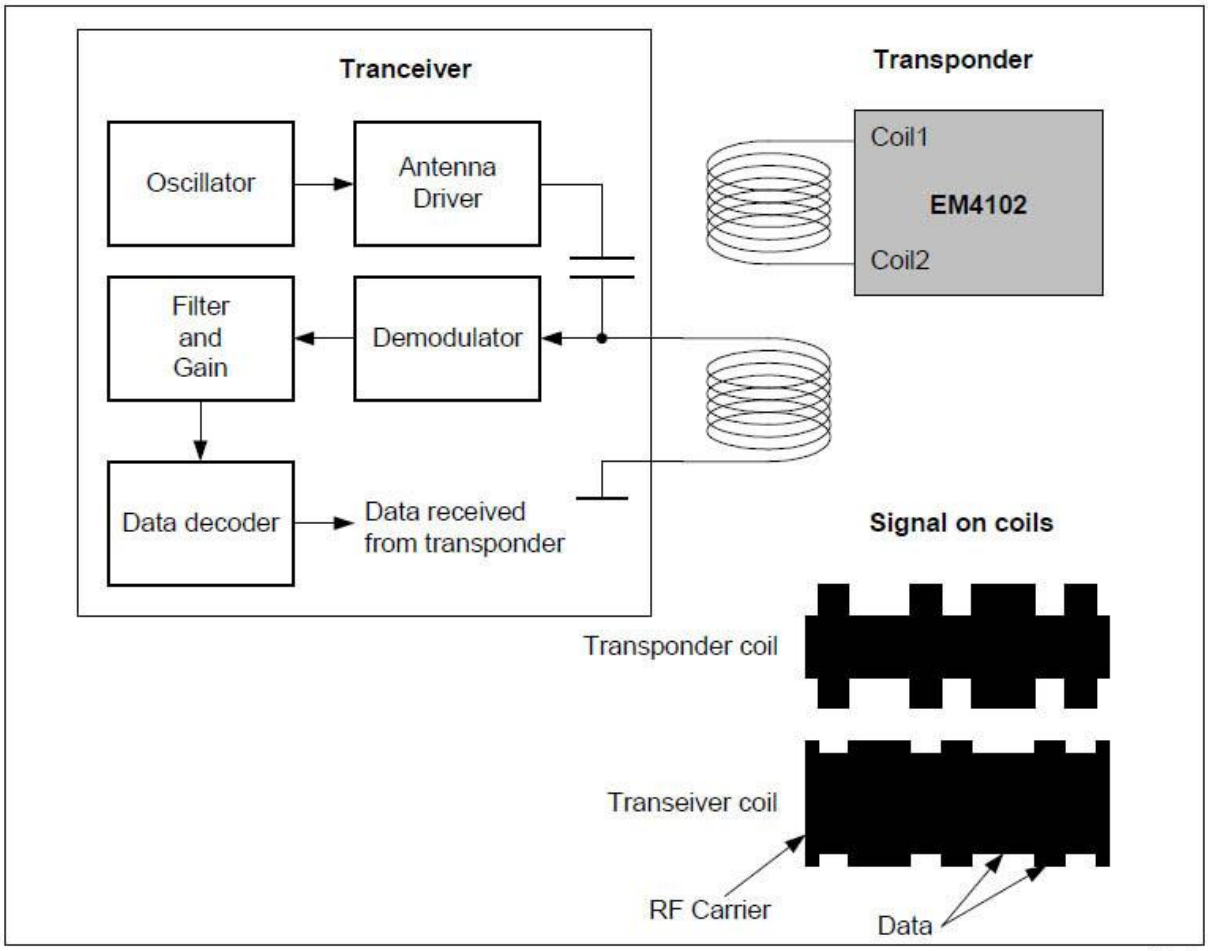


Fig. 3

Electrical Characteristics

V_{DD} = 1.5V, V_{SS} = 0V, f_{C1} = 134kHz square wave, T_a = 25°C
V_{C1} = 1.0V with positive peak at V_{DD} and negative peak at V_{DD} -1V unless otherwise specified

Parameter	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Supply Voltage	V_{DD}		1.5		1)	V
Rectified Supply Voltage	V_{DDREC}	V_{COIL1} - V_{COIL2} = 2.8 VDC Modulator switch = "ON"	1.5			V
Coil1 - Coil2 Capacitance	C_{res}	V_{coil}=100mVRMS f=10kHz		78 2)		pF
Power Supply Capacitor	C_{sup}			125		pF
Manchester and biphase versions						
Supply Current	I_{DD}			0.6	1.5	µA
C2 pad Modulator ON voltage drop	V_{ONC2}	V_{DD}=5.0V I_{VDDC2}=1mA with ref. to V_{DD}	150	220	280	mV
PSK version						
Supply Current	I_{DDPSK}			0.9	2.0	µA
C2 pad Modulator ON voltage drop	$V_{ONC2PSK}$	V_{DD}=5.0V I_{VDDC2}=100µA with ref. to V_{DD}	500	650	800	mV

Note 1) The maximum voltage is defined by forcing 10mA on COIL1 - COIL2
Note 2) The tolerance of the resonant capacitor is ± 15% over the whole production. On a wafer basis, the tolerance is ± 2%

Timing Characteristics

V_{DD} = 1.5V, V_{SS} = 0V, f_{coil} = 134kHz square wave, T_a = 25°C
V_{C1} = 1.0V with positive peak at V_{DD} and negative peak at V_{DD} -1V unless otherwise specified
Timings are derived from the field frequency and are specified as a number of RF periods.

Parameter	Symbol	Test Conditions	Value	Units
Read Bit Period	T_{rdb}	depending on option	64, 32, 16	RF periods

Timing Waveforms

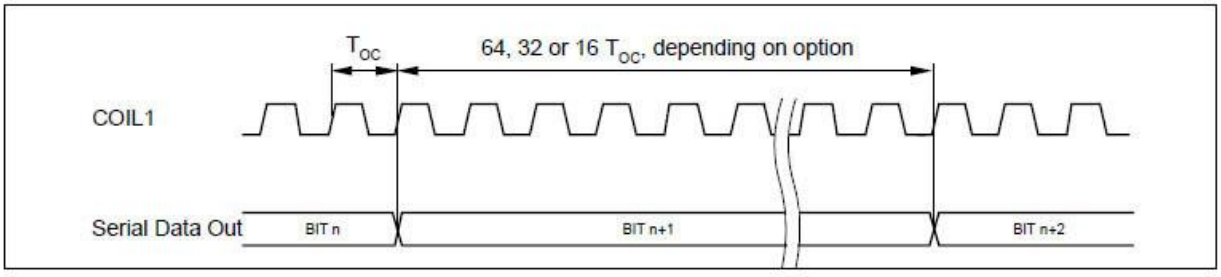


Fig. 4

Block Diagram

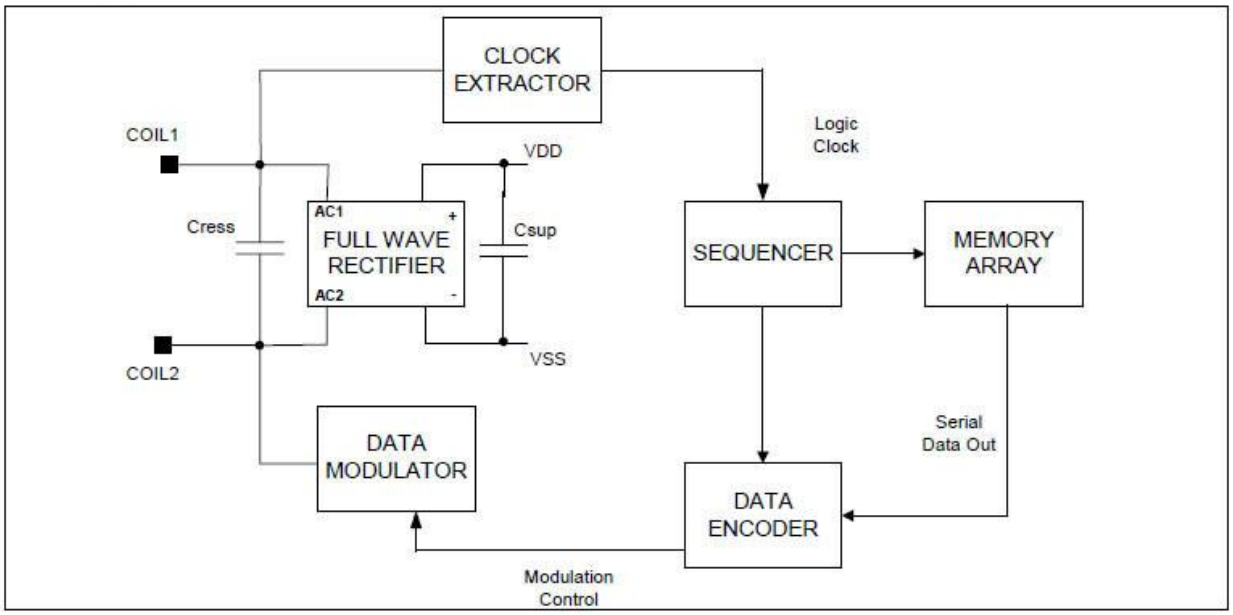


Fig. 5

Functional Description

General

The EM4102 is supplied by means of an electromagnetic field induced on the attached coil. The AC voltage is rectified in order to provide a DC internal supply voltage. When the last bit is sent, the chip will continue with the first bit until the power goes off.

Full Wave Rectifier

The AC input induced in the external coil by an incident magnetic field is rectified by a Graetz bridge. The bridge will limit the internal DC voltage to avoid malfunction in strong fields.

Clock Extractor

One of the coil terminals (COIL1) is used to generate the master clock for the logic function. The output of the clock extractor drives a sequencer.

Sequencer

The sequencer provides all necessary signals to address the memory array and to encode the serial data out.
Three mask programmed encoding versions of logic are available. These three encoding types are Manchester, biphase and PSK. The bit rate for the first and the second type can be 64 or 32 periods of the field frequency. For the PSK version, the bit rate is 16.
The sequencer receives its clock from the COIL1 clock extractor and generates every internal signal controlling the memory and the data encoder logic.

Data Modulator

The data modulator is controlled by the signal Modulation Control in order to induce a high current in the coil. The coil 2 transistor drives this high current. This will affect the magnetic field according to the data stored in the memory array.

Memory Array for Manchester & Bi-Phase encoding ICs

The EM4102 contains 64 bits divided in five groups of information. 9 bits are used for the header, 10 row parity bits (P0-P9), 4 column parity bits (PC0-PC3), 40 data bits (D00-D93), and 1 stop bit set to logic 0.

1	1	1	1	1	1	1	1	1	9 header bits
8 version bits or customer ID				D00	D01	D02	D03	P0	
				D10	D11	D12	D13	P1	
				D20	D21	D22	D23	P2	
32 data bits				D30	D31	D32	D33	P3	
				D40	D41	D42	D43	P4	
				D50	D51	D52	D53	P5	
				D60	D61	D62	D63	P6	
				D70	D71	D72	D73	P7	
				D80	D81	D82	D83	P8	
				D90	D91	D92	D93	P9	10 line parity bits
				PC0	PC1	PC2	PC3	S0	

4 column parity bits

The header is composed of the 9 first bits which are all programmed to "1". Due to the data and parity organisation, this sequence cannot be reproduced in the data string. The header is followed by 10 groups of 4 data bits allowing 100 billion combinations and 1 even row parity bit. Then, the last group consists of 4 event column parity bits without row parity bit. S0 is a stop bit which is written to "0"
Bits D00 to D03 and bits D10 to D13 are customer specific identification.
These 64 bits are outputted serially in order to control the modulator. When the 64 bits data string is outputted, the output sequence is repeated continuously until power goes off.

Memory Array for PSK encoding ICs
The PSK coded IC's are programmed with odd parity for P0 and P1 and always with a logic zero.
The parity bits from P2 to P9 are even.
The column parity PC0 to PC3 are calculated including the version bits and are even parity bits.

Code Description
Manchester
There is always a transition from ON to OFF or from OFF to ON in the middle of bit period. At the transition from logic bit "1" to logic bit "0" or logic bit "0" to logic bit "1" the phase change. Value high of data stream presented below modulator switch OFF, low represents switch ON (see Fig. 6).

Biphase Code
At the beginning of each bit, a transition will occur. A logic bit "1" will keep its state for the whole bit duration and a logic bit "0" will show a transition in the middle of the bit duration (see Fig. 7).

PSK Code
Modulation switch goes ON and OFF alternately every period of carrier frequency. When a phase shift occurs, a logical "0" is read from the memory. If no shift phase occurs after a data rate cycle, a logical "1" is read (see Fig. 8).

Manchester Code

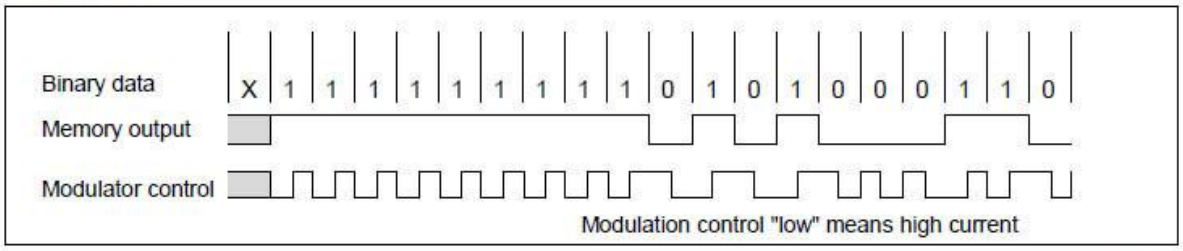


Fig. 6

Biphase Code

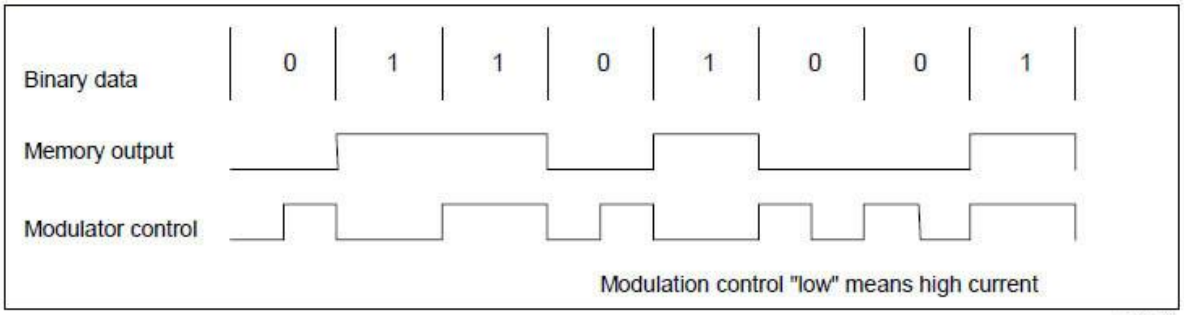


Fig. 7

PSK Code

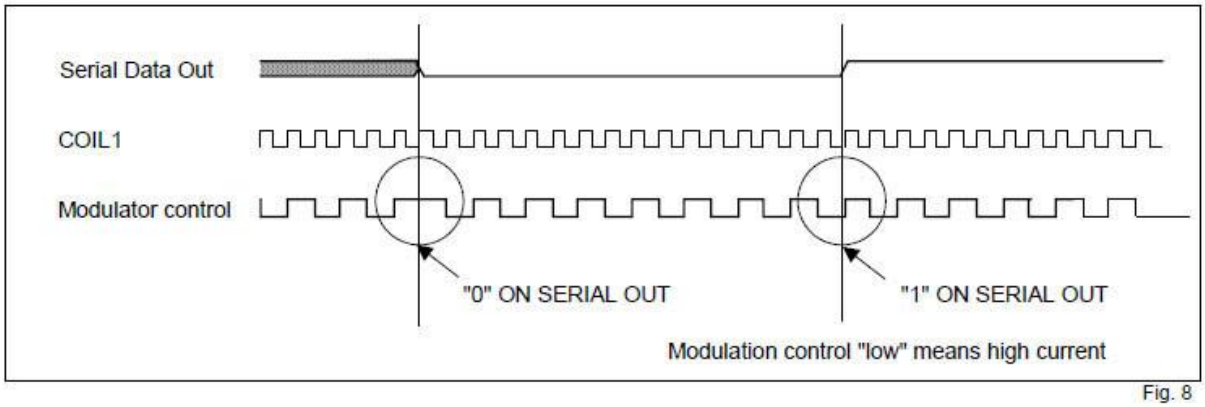
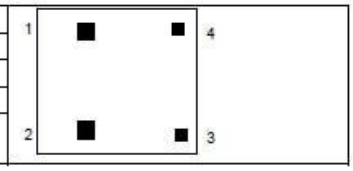


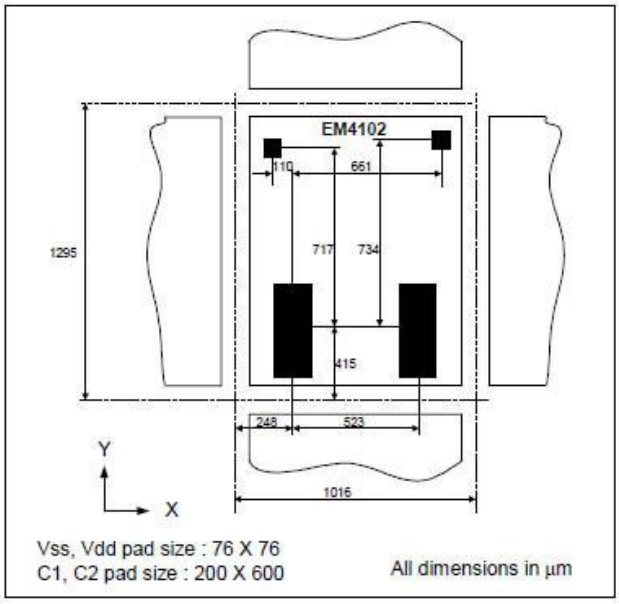
Fig. 8

Pad Description

Pad	Name	Function
1	COIL2	Coil terminal 2 / data output
2	COIL1	Coil terminal 1 / clock input
3	VDD	Positive internal supply voltage
4	VSS	Negative internal supply voltage



CHIP Dimensions



Vss, Vdd pad size : 76 X 76
C1, C2 pad size : 200 X 600

All dimensions in µm

Ordering Information

Die Form

EM4102 A6 WS 11 E - %%%

Version:
A6 = Manchester, 64 clocks per bit
A5 = Manchester, 32 clocks per bit
B6 = Bi-Phase, 64 clocks per bit
B5 = Bi-Phase, 32 clocks per bit
C4 = PSK, 16 clocks per bit

Die form:
WW = Wafer
WS = Sawn Wafer/Frame
WT = Sticky Tape
WP = Waffle Pack (note 1)

Thickness:
7 = 7 mils (178um)
11 = 11 mils (280um)

Bumping:
" " (blank) = no bumps
E = with Gold Bumps

Customer Version:
%%% = only for custom specific version

This chart shows general offering; for detailed Part Number to order, please see the table "Standard Versions" below.

Card Form

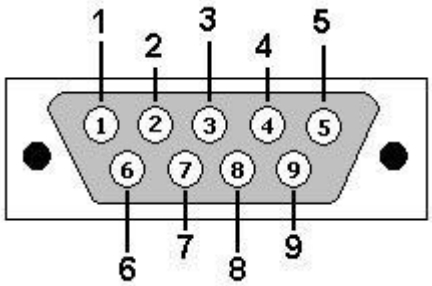
This chart shows general offering; for detailed Part Number to order, please see the table "Standard Versions" below.

EM4102 A6 CX - %%%

Version:
A6 = Manchester, 64 clocks per bit
A5 = Manchester, 32 clocks per bit
B6 = Bi-Phase, 64 clocks per bit
B5 = Bi-Phase, 32 clocks per bit
C4 = PSK, 16 clocks per bit

Card form:
CX = Card without magnetic strip, GLOSS
CY = Card without magnetic strip, MATT
CM = Card with magnetic strip, GLOSS
CN = Card with magnetic strip, MATT

Customer Version:
%%% = only for custom specific version



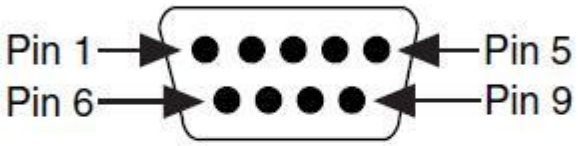
DB9 RS232 Male

RS232 Pin Assignments
(DE9 PC signal set)

Pin 1	Received Line Signal Detector (Data Carrier Detect)
Pin 2	Received Data
Pin 3	Transmit Data
Pin 4	Data Terminal Ready
Pin 5	Signal Ground
Pin 6	Data Set Ready
Pin 7	Request To Send
Pin 8	Clear To Send
Pin 9	Ring Indicator

The connector on the PC has male pins, therefore the mating cable needs to terminate in a DE9/F (Female pin) connector.

DE-9 Male



Pin	232 DTE	232 DCE	422/485
1	DCD[2]	DCD	GND
2	RXD	TXD	CTS+ (HSI+)
3	TXD	RXD	RTS+ (HSO+)
4	DTR[2]	DSR	RXD+
5	GND	GND	RXD–
6	DSR[2]	DTR	CTS– (HSI–)
7	RTS	CTS	RTS– (HSO–)
8	CTS	RTS	TXD+
9	RI[2]	RI	TXD–

Note: DCE mode supported on USB-232/2 and USB-232/4 only.

[2] These signals are "No Connect" on the PCI-232I and PXI-8422 ports and ports 9–16 on legacy 16-port boards.

GSM w/ GPRS modem